# Ensemble of Data Augmentation Techniques for Efficient Augmentation in NLP

Nandan Parmar [1]
[1]*Department of Computer Science Engineering, Symbiosis Institute of Technology, Pune India*

***Abstract*—In the last decade, NLP has made significant advances in machine learning. In so many machine learning scenarios, there isn't enough data available to train a good classifier. Data augmentation can indeed be utilized to solve this problem. It utilizes transformations to artificially increase the amount of available training data. Due of linguistic data's discrete character, this topic is still relatively underexplored, in spite of the huge rise in usage. A major goal of the DA techniques is to increase the diversity of training data, allowing the model to better generalize when faced with novel testing data. This study uses the term "data augmentation" to allude as a broad concept that encompasses techniques for transforming training data. While most text data augmentation research focuses on the long-term aim of developing end-to-end learning solutions, this study focuses on using pragmatic, robust, scalable, and easy-to-implement data augmentation techniques comparable to those used in computer vision. In natural language processing, simple but successful data augmentation procedures have been implemented and inspired by such efforts, we construct and compare ensemble data augmentation for NLP classification. We are proposing an ensembling of simple yet effective data augmentation techniques. Through experiments on various dataset from kaggle, we show that ensembling of augmentation can boost performance with any text embedding technique particularly for small training sets. We conclude by carrying out experiments on a classification datasets. Based on the results, we draw conclusion that Effective DA approach by ensembles of data augmentation can help practitioners choose suitable augmentation technique in different settings.**

***Index Terms*—Text Data Augmentation, NLP, Class Imbalance, Text Embeddings**

## I. INTRODUCTION

The Text messaging has long been the most common method of communication. Online social networks have given birth to a larger variety of textual data: (OSN). Allegiance and political views are only two examples of the enormous range of information that OSN users utilize to express themselves through the medium of text. Text may be improved in many ways, including its information quality and readability, using computer science techniques such as word placement analysis and other syntactic and semantic aspects as well as calculating term frequencies. All computer text processing and analysis approaches are based on Natural Language Processing (NLP) [1].

Text information is used in Machine Learning (ML) to categorize fresh text input into one or more unique classes. The practice of categorizing or organizing text data into classes is known as text classification [2]. It is an essential component of Natural Language Processing. NLP consists of a variety of tasks, from text categorization to question answering, but regardless of what we do, the amount of data we must train in our model has a significant influence on the model's performance, more data we have, the better results we can achieve and because there have been an increasing ton of good models in recent years. More and more data are required for a better model, the problem of inadequate data is frequently encountered in the industrial internet. What can we do to increase the size of your dataset, probably more data is a simple option, but gathering and categorizing more observations, can be a costly and time-consuming task so this arises the question of what some alternatives are and hence the Use data augmentation can improve the quality of your text data [3]. However, an increase in training data does not always imply that the learning problem has been solved. Nevertheless, the quality of a supervised classifier is still determined by the data.

Data Augmentation (DA) is a technique that allows us to artificially enhance the quantity of training data by synthesizing different versions of actual datasets without collecting the data and instead using the data we already have. This can be used with any type of data, including number, pictures and speech. To improve performance in the

classification task, the data must be modified to preserve the class categories. We employ data augmentation to lessen dependency on training data preparation and enhance the development of more accurate machine learning models. Data augmentation is a term that comes from the field of computer vision and refers to a variety of approaches for artificially creating such data. Transformations such as rotations, converting image to grayscale or changes in the RGB channel are appropriate for pictures, as the model should be consistent for these. Similarly, In Speech recognition employs methods that alter the sound or speed of speech, such as to increase the sound dataset we can pitch up or lower down the audio sample.

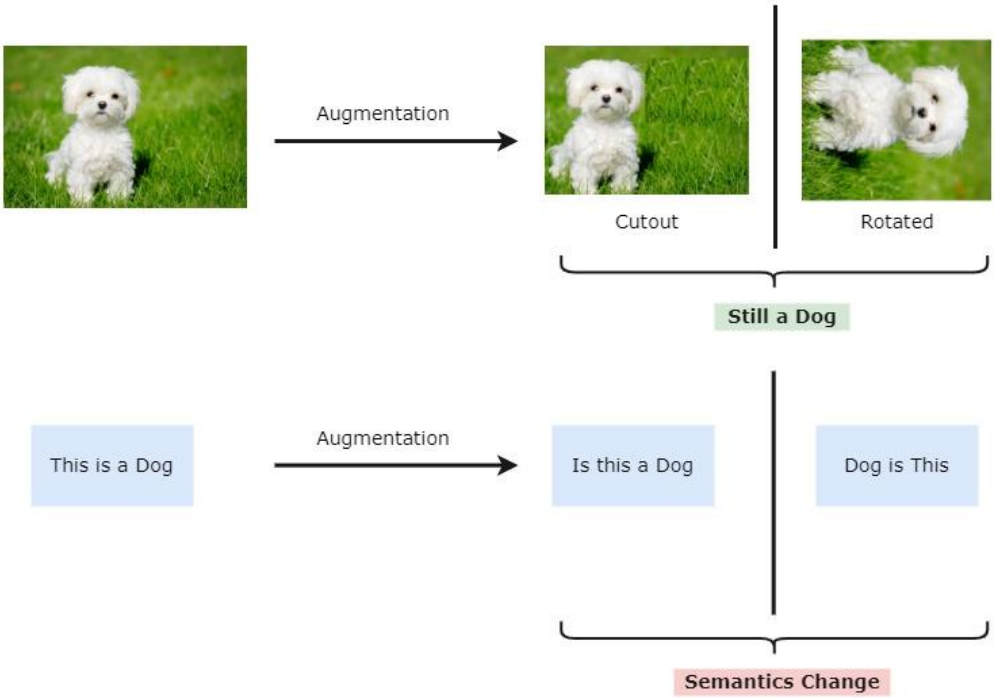


Fig. 1. Challenge of Semantic Transformation in NLP.

In contrast to that data augmentation research in Natural Language Processing (NLP) faces the challenge of developing uniform rules for textual data changes that can be automatically created while maintaining labeling quality. Because natural language is discrete, applying DA techniques to it is more challenging and underexplored in NLP.

As a result, there is a rising interest in employing data augmentation approaches at the token and sentence level in natural language processing (NLP) tasks, such as text classification [4], natural language inference [5] and machine translation [6]. Textual data augmentation can come in many different forms and numerous DA approaches have lately been proposed, but the challenge still exists, and it is being tackled by many researchers in many research directions. As interest in and work on this area rising, now is an appropriate time for a paper like ours to propose effective data augmentation for NLP applications. By researching data augmentation strategies, we fill in the gap by developing a solution that takes into account the class imbalance and works on any text dataset for natural language processing (NLP) tasks in this research.

The contribution of this paper is to:

- To adopt the concept of data augmentation as a major concept, encompassing some techniques together that lets transformation of training data.
- To show that by creating new and additional data can enlarge the quality of a solution and improves the accuracy of machine learning models to train datasets.

## II. LITERATURE SURVEY

In the following section Various data augmentation approaches for text data are summarized, described, and divided into multiple groups. The techniques are typically presented in a logical sequence for the specific group. We've divided them into six groups. Techniques that are applied for text classification

are included mostly, while augmentation techniques for other textual tasks are mentioned if they match the category Feng et al. [7] surveys, which provide a bird's eye perspective of DA in NLP. Liu et al. [8] presented a broad overview of NLP augmentations, but they both divided the categories into groups based on the methodologies. As a result, these groupings are either too narrow or too broad. However, Bayer et al. [9] published a survey on DA that focuses solely on text categorization.

In this section we'll let you have a comprehensive introduction of NLP's DA approaches. One of our primary goals is to explain how data augmentation works and why it is effective. To make this easier, we group DA techniques based on the variety of augmented data, as increasing the diversity of training data is one of the main aims of DA. Several ways for applying Data Augmentation to text data are presented in this brief overview. These enhancements are categorized as Lexical Based, Model Based, Rule Based, Noising Based, Injecting Based, and Sentence Based.

Later in this section we differentiate the augmentation techniques into two structural categories. we give a swift overview about the performance of the data augmentation techniques. These improvement indications would give a swift overview on how well the technique can perform. For a comprehensive perspective, datasets are also given. Although in-depth details must be taken from the papers themselves, it will give a more comprehensive picture. This section also analyzes where these given augmentation techniques have been used in NLP tasks and what are their applications in NLP.

We'll now go through various DA approaches that are applicable to NLP tasks and six different types of data augmentation strategies in the sections below.

2.1. Injecting Error Techniques

Injecting errors into input data is one of the simplest data augmentation techniques, especially on a character level. Even though most applications come with word correction, there are still typos in chatbots. To get around this, we may let our model "see" the probable outcomes before making an online prediction.

2.1.1. Keyboard Error Injection

The basic idea of this method is to replace one letter with a distant letter on the keyboard. The errors are inserted as an artificial noise based on keyboard distance. Belinkovand Bisk [10] defined this technique and added artificial noise to their training data for their NMT task, they got varied BLEU scores with their artificial noise methods do not revise any of the current designations.

2.1.2. OCR Error Injection

Basic idea of the technique is to inject noise in recognizing characters, in other words to replace the text with a possible OCR error by leveraging pre-defined OCR mapping. For example, ‘O’ can be replaced by ‘0’ (zero), similarly ‘I’ can be replaced by ‘1’ (one). A.M. Ningtyas et al. [11] used OCR error injection for their character augmentation for their Medical Concept Normalization and Named Entity Recognition task.

2.2. Noising-Based Techniques

The noising-based techniques bring a small amount of noise that has no effect on the semantics, causing it to differ from the original text correctly. Through knowledge of language characteristics and prior information, humans can considerably minimize the effects of faint noise on semantic understanding, but this noise may provide difficulties to the model. Aside from increasing training data availability, it also improves model resilience as a result of this method

2.2.1. Unigram Noising

The basic idea of this technique is to replace words sampled from the certain probability frequency distribution. The frequency distribution is based on how frequently each word occurs in training data. Xie et al. [12] used this method and by addition of noise they were able to improve the quality of classifiers.

2.2.2. Blank Noising

This method also was proposed by Xie et al. [12] where the idea was to replace words with ‘_’ (underscore), as a placeholder token. They used this technique to avoid the overfitting problem on

specific contexts which in turn gave them improved results in BLEU scores.

2.2.3. Random Swapping

Wei et al. [13] devised a method in which they randomly selected two words from a phrase and then switched their locations. n times the length of the statement, this strategy was applied. When they were doing this, Dai et al. [14] also and his colleagues decided to split the collection of tokens into segments based on their labels, and then they shuffled the tokens inside those segments without affecting the order of the tokens.

2.2.4. Random Deletion

Another EDA approach used by Wei et al. [13] is to eliminate words from a phrase at random with probability p, which also a easy methodology. They saw an improvement in the classifier's performance. For their Spoken Language Understanding tasks, Peng et al. [15] employed this strategy to enrich input dialogue acts by removing slot values.

2.2.5. wordMixup

Guo et al. [16] proposed this noising-based technique where for wordMixup, the sequences should be zero padded to maintain the same dimensions, the generated word embedding is then sent through the typical text classification loop. In the given proportion, the cross-entropy was calculated for both the labels of the actual text. They used wordMixup to interpolate a new sample, and the interpolated label is as follows:

Where N words in sentence can be represented as a matrix $B \in RN \times d$, $B^i_t$, $B^j_t$,One word belongs to each row t of the matrix denoted by Bt. The embedding vectors of the input sequences are $B_i$and$B_j$, while the appropriate class labels of the data are $y^i$ and $y^j$, then, the new sample ($\tilde{B}^{ij}$; $\tilde{y}^{ij}$) is used for training.

$$\widetilde{B}_t^{ij} = \lambda B_t^i + (1-\lambda) B_t^j$$

$$\widetilde{y}^{ij} = \lambda y^i + (1-\lambda) y^j$$

Mixup is also introduced into NER by Chen et al. [17] who propose both Intra-LADA and InterLADA.

2.3. Lexical Substitution Techniques

Lexical Substitution alter words and phrases in the text to create enhanced text while ideally preserving the original text's semantic meaning and labels. Basically, a replacement is done in this type of category

2.3.1. Synonym Replacement

Synonym Replacement is the most chosen data augmentation technique, In this technique an arbitrary word is taken from sentence and it gets replaced with its true synonym, while keeping the semantics of certain text unaffected. For example, a thesaurus substitution with Wordnet database that contain lexical triplets or words and because it is a manually organized database with relations between words it is used as external resources.

Zhang et al. [18] used this technique first off and used the synonym-based substitution derived from Wordnet, for text classification tasks. Synonym is chosen on the basis of geometric distribution, in other words, synonym is chosen on based on the sorted similarity of the certain word. Wei and Zou [13] proposed Easy Data Augmentation (EDA), where they also replace original words with their synonyms applying Wordnet. But instead of choosing synonym based on geometric distribution they choose n words randomly from sentence which get replaced by random synonym.

2.3.2. Word Embeddings Replacement

Replaces words in pre-trained word vectors with the nearest neighbor words in the embedding space in the vector space using this approach. Wang et al [19] used the technique Word-Embeddings Substitution, where they categorize annoying behaviors using Tweets, they augment tweets needed to learn a topic model to better classify annoying tweets. They replaced original word by utilizing k-nearest-neighbor words using cosine similarity.

Liu et al. [20] solely utilized word embeddings to regain synonyms and Marivate and Sefara [21] used random replacement selection with probability proportional to cosine similarity and Rizos et al. [22] have utilized embedding substitution on cosine similarity threshold and POStag matching.

2.3.3. Antonym Replacement

Like Synonym replacement, the antonyms of a word can be used in the same vicinity. It also applies semantic meaning for the word, means the noun or verb can replaced by its antonym. This technique reverses the sentence meaning. Haralabopoulos et al. [23] proposed this method and utilized augmented data for the reversion of the classification where they achieved improvement in classification accuracy by 0.35%.

Kashefi and Hwa [24] used this method to solve the problem of best suitable heuristic and data augmentation for classification tasks. Madukwe et al. [25] used this technique for their hate speech detection task by replacing word its antonym because it is not considered a semantically invariant transformation [26].

2.4. Rule-Based Techniques

This group of technique takes some natural language heuristics to ensure the preservation of sentence semantics.

2.4.1. Text Surface Transformation

This Technique rely on heuristics that make sure to maintain the sentence semantics, Coulombe et al. [26] introduced these patterns matching transformation using regular expressions without changing its semantics. They transformed verbs, modal and negation from contraction to expansion conversely. In the similar way Regina et al. [27] used this technique by relying on word pair dictionaries, they transformed verbal forms by expansion, inversely among set of words to get the replacements.

2.4.2. Dependency Tree Manipulation

Coulombe et al. [26] also introduced sentence-level rule-based manipulation technique where the aim is to analyze and construct the original sentence's dependency tree, then change it using rules to generate a paraphrased statement, this transformed dependency tree is nothing but augmented data. Basically, transformation from active to passive voice and vice versa, Louvan et al. [28] also used this technique to generate smaller sentences by cropping fragments on syntax tree.

2.5. Model-Based Techniques

In this category are techniques that use pre-trained models or transformer models to generate augmented data.

2.5.1. Masked Language Models

Masked Language Transformer models for-instance They have learned to anticipate masked words based on the context surrounding them, which has been pretrained with many pretext tasks. In this approach, the problem of ambiguity is lessened since context semantics is considered. Palomino et al. [29] used Masked Language Model technique to generate new sentences by masking words in a sentence so that these masks can bring out more diverse sentences.

Kobayashi [30] used this data augmentation to replace word with other words that predicted by a MLM at those word positions, he altered the MLMs to integrate the label in the model for word prediction while Wu et al. [31] transformed the BERT language model architecture and called it (c-BERT) that is label conditional language model and further Jiao et al. [32] used this technique where they apply the tokenizer of BERT and found out that the data generated by BERT is low while tokenizing words into multiple word pieces that is why authors proposed GloVe embeddings substitution so that they replace the word piece if word is a complete word by masking and using BERT to predict such words.

2.5.2. Seq2Seq Model

An attempt is made to create more diverse phrases by using duplication-aware attention and diverse-oriented regularization in this data augmentation. The Seq2Seq model learns the internal mapping between target and source distributions. Seq2Seq data augmentation was proposed by Hou et al. [33] for task-based conversation systems' language understanding module. They use the Seq2Seq model to generate a new utterance by feeding the delexicalized input phrase and the supplied variable rank as input. The concatenated multiple input expression is represented by Hou et al. [33] using an L-layer transformer. For each label, Kang et al. [34] utilized a Seq2Seq model to train and then used the Seq2Seq model to develop the augmented data for a text.

2.6. Sentence Level Transformation

This grouping majorly focuses on techniques that are on sentence level, even though Dependency Tree Manipulation and Seq2Seq Model based augmentation techniques also falls into this category but they both were required to group into different category because of their augmenting behavior. These types of augmentation are performing well in maintaining labels.

2.6.1. Backtranslation

Using this method, the text is first translated into another language, and then translated back into the original language, in the reverse order. This approach is effective because it ensures correct grammar and labels are preserved. In comparison to other approaches back translation rewrites the whole sentence and does not directly replace individual words as you can see in figure 2.16. When Sennrich et al. [35] used this technique to increase the quality of their Neural Machine Translation model, they included training phrases from monolingual target languages in their data. In order to create synthetic training data, Yu et al. [36] utilized English to French neural machine translation.

Xie et al. [37] used this strategy to enrich the unlabeled text in order to create a semi-supervised model on 20 labeled instances that beat a standard model trained on 25,000 labeled examples, according to the results published in Nature.

2.6.2. Swapped Crossover Augmentation

In this augmentation method the text is swapped on sentence level. In this approach the text is splitted into two halves then the halves of two random text with the same polarity (positive/negative) are swapped. Even though the data generated in this manner is grammatically incorrect and semantically unsound, yet it contains more semantic information and emotional polarity than a single word. Luque [38] proposed this approach in his paper on sentiment analysis. It is based on the chromosomal crossover procedure used in genetics. The author had no effect on the accuracy, but it did assist with the F1 score in the paper.

$$\tilde{B}^{ij}_{\{k\}} = f(B^i)_{\{k\}} + (1-\lambda) f(B^j)_{\{k\}}$$
$$\tilde{y}^{ij} = \lambda y^i + (1-\lambda) y^j$$

After sentence embeddings the embedding vectorB̃ij will be passed on to Softmax layer. Si et al [39] employ a Mixup technique for text classification.

2.6.3. sentMixUpAugmentation

Guo et al. [16] also proposed sentence level mixup technique where the hidden embeddings created are of the same length, The word embeddings are sent through an LSTM/CNN encoder, and the final hidden state is used as the sentence embedding. These embeddings are mixed in a certain proportion before being sent to the final classification layer. The cross-entropy loss is estimated using both original sentence labels in the proportion given. In this approach, the set of sentences $B^i$and$B^j$are encoded first into set of sentence embeddings $f(B^i)$ and $f(B^j)$, where f is the sentence encoder, mixup was carried in for each kth level of the sentence embedding as shown below:

2.7. Overview of DA Techniques

Before going on to review where some of these data augmentation techniques are applied on which NLP tasks, first we attempted to summarize the most important information in the form of tables.

We also attempt to gather information on the improvements. The following table should provide a quick overview of how well this strategy has performed on certain NLP tasks. Table 1 summarizes the NLP task where this augmentation has been used, like which augmentation strategy has been used on what kind of datasets with improvements of the various techniques and also we try to give out the advantages as well as the certain limitations of those augmentation techniques. Various authors used these techniques in their papers for several NLP purposes.

TABLE I. Overview of DA Techniques

| Authors | Augmentation Strategy | Tasks | Datasets | Advantages | Limitations | Improvements (on base) |
|---|---|---|---|---|---|---|
| [13] | Keyboard Error Based | Sentiment Prediction | IMDB movie reviews | Vigorous Models | Varying Accuracy | 84.02% |
| [6] | OCR Error Based | Named Entity Recognition | CADEC,PsyTAR | Vigorous Models | Injecting information may alter the original label | +0.04 accuracy on PsyTAR |
| [7] | Unigram Noising | Machine translation | IWSLT 2015 | Improve model sturdiness. | Imprecise syntax and semantics. | +0.9 BLEU score |
| [7] | Blank Noising | Language Modeling | Penn Treebank, Text8 corpus | Improve model sturdiness. | Limited variety in single method. | +0.7 BLEU score |
| [8] | Random Swapping | Text classification | SUBJ, TREC | Improved Performance in few Classifiers | Labels are not preserving | 0.2 performance gain |
| [8] | Random Delete | Sentiment Analysis | SST, CR | Improved Performance in few Classifiers | Unclear syntax | 0.1 performance gain |
| [9] | Synonym Substitution | Text classification | AG's News, DBPedia, Yelp | Simple to use. | The range and POS of substitution words are limited. Substitutions are narrowed by a database, like WordNet | +1.36 on Yelp |
| [10] | Antonym Substitution | Text classification | SemEval, Crowd | Simple to use. | The sentence semantics may be affected if too many replacements occur. | +0.35% average accuracy On SemEval & Crowd |
| [11] | Word-Embeddings Substitution | Text classification | Petpeeve dataset | Higher substitution rate and wider substitution range. | Technique cannot resolve the problem of uncertainty | +2.4% on F1 score on Petpeeve |
| [12] | Transformer Model Based | Text classification Sequence labeling | SST, SUBJ | Technique improves the problem of ambiguity | Limited to the word level. | +1.5 accuracy on SST-2 |
| [13] | Text Surface Transformation | Sentiment Analysis | IMDB movie reviews | Technique preserves the original sentence semantics. | This method requires artificial heuristics | +1.37% on accuracy |

| | | | | | | |
|---|---|---|---|---|---|---|
| [13] | Syntax-tree Manipulation | Sentiment Analysis | IMDB movie reviews | Technique preserves the original sentence semantics | There is little coverage and variety. | +0.77% accuracy |
| [14] | Back Translation | Machine Translation<br><br>Dialogue generation | WMT '15 (en-de), IWSLT '15 (en-trl). Yelp-5 | Robust applicability.<br><br>Technique ensures proper grammar and unaffected semantics | Because of fixed-machine translation models, there is little controllability and diversity. | +1.65 on Yelp-5 |
| [15] | Seq2Seq Model Based | Adversarial Example Generation | SciTail, SNLI | Sturdy variety And application | Require training data and High level of training difficulty | +2.38 increase in F-score |
| I | MixUp for Text | Semantic Parsing<br><br>Language modeling | Trec, SST-1, SST-2 | Creates augmented data by combining different labels. | Less comprehensible and more difficult to train | +1.61 accuracy on tree dataset |
| [17] | Conditional Generation | Text classification<br><br>Question answering | ATIS, TREC, WVA | Appropriate for data-sparse scenarios | Required for unlabeled data and<br><br>Poor application | +1.8% accuracy on TREC dataset |

Several DA techniques have evolved in NLP in recent times, comparing their effectiveness is challenging because different types of NLP tasks, evaluation metrics, datasets, and model architectures are employed, but here are NLP tasks wherein DA is used to improve data variety in Table 2.

TABLE II. Applications of Augmentation Techniques.

| Authors | Application | Augmentation Strategy | Highlights |
|---|---|---|---|
| [41] | Machine Translation | Back-Translation | By augmenting existing sentences in the corpora, the training data is increased. |
| [42] | Text Summarization | Iterative back-translation | Iterative backtranslation technique for the German language that employs synthetic data in addition to actual summarizing data |
| [36] | Question Answering | Back-Translation | Increase the number of training data for any language-based activity, including reading comprehension. |
| [43] | Automated Augmentation<br><br>&Dialogue Generation | Masked Language Model &<br><br>Backward translation | A trainable data manipulation model for augmenting effective training samples and lowering the weights of inadequate samples. |
| [44] | Mitigating | Counterfactual | To augment the data, removing gender bias in text by changing |

| | Bias | Data augmentation | pronouns ("he" becomes "she"). |
|---|---|---|---|
| [45] | Visual Question Answering | Semantic annotations&<br>Conditional Generation | They used LSTM to generate question and answer pairs for the Visual Question Answering (VQA) task. |
| [46] | Multimodal | Conditional Generation | They proposed a strategy for augmenting training data by substituting source data samples with shorter overlapping samples derived from them. |
| [47] | Dialogue Processing | Random noise Injection | In dialogue processing, spoken language understanding, attempts were made to increase SLU performance by data augmentation. |
| [48] | Sequence Tagging | Conditional generation with LM | A two-step data augmentation process. Firstly a LM is trained over sequences of tags and words linearized according to a certain scheme. Secondly, sequences from this language model are sampled and de-linearized to produce new instances. |
| [49] | Grammatical Error Correction | Noise Injection | By altering latent representations of grammatical sentences, noise is injected into the latent representation of a sentence to synthesize new samples. |
| [50] | Question Classification<br>Sentiment Analysis | Conditional Generation | They proposed a new data augmentation strategy that, through an active learning process, it can boost the effectiveness of text classifiers.<br>Transforms the data generation task into an optimization problem that enhances the output's usefulness. |

Numerous data augmentation techniques have been presented, however based on how the methods have been utilized and how they have affected NLP tasks, we divide the data augmentation techniques into linguistic and non-linguistic divisions. By linguistic characteristic, we imply that the meaning is kept after data augmentation and that the enhanced data adheres to the correct linguistic form. Techniques like synonym replacement, word embeddings substitution, Backtranslation, Model based techniques comes under Linguistic property. In the nonlinguistic group, augmentation is achieved using techniques such as injecting and noising.

TABLE III. Categorization of Augmentation Techniques.

| Sr. No. | Linguistic | Non-linguistic |
|---|---|---|
| 1 | Synonym Replacement | Random insertion |
| 2 | Word-Embeddings Substitution | Random swap |
| 3 | Masked Language Model | Random deletion |
| 4 | Back-Translation | Noise injection |

2.8. Limitations of Previous Studies

There are many data augmentation techniques out there but still there are some limitations regarding those methods, here down below are listed few limitations of previous studies that we are trying to improvise:

- Traditional Data augmentation methods does not perform well in a generalized NLP Task.
- Traditional Augmentation Strategies are not much effective in augmenting corpus.
- Traditional methods usually employed a single baseline model for classification and discrimination.
- Few Augmentation strategies could not improve the classification accuracy because of generalization problem during text embeddings.

2.9. Applications of DA

- Enhancing the accuracy of model prediction by
  - Incorporating additional training data into the classifiers
  - Preventing data scarcity to improve models.
  - Minimizing data overfitting.
  - Enhancing the model's generalization capabilities while assisting in the resolution of class imbalance problems in classification.
- Cutting the cost of data gathering and labelling the data.
- Allows the prediction of unlikely events.
- Prevents issues with data privacy.

## III. THEORETICAL BACKGROUND

3.1. Datasets

In this section we will give the overview of the datasets we are using for our research purpose, we are using three datasets for this experimental work, a brief dataset description of each dataset is given below.

3.1.1. Spam Email Dataset

Spam is an unwanted or undesired text message; if that message is an e-mail, it is referred to as spam e-mail. Spam is neither a virus or a danger that may harm a computer, although it can cause problems if massive amounts of spam are transmitted. Sometimes it is very difficult to filter out the legitimate mails out of spam mails. This dataset we have downloaded and used from Kaggle which has 1082 rows containing spam and ham mails. This a small size dataset. In the dataset there is serial number determining the no of mails, there is another column of Message body contains the actual text followed by the label of Spam and Ham which we have later converted into 0 and 1 for label preservation. In this case 1 is the Spam mail and 0 is the legitimate mail.

3.1.2. Yelp Coffee Reviews Dataset

Reviews can impact customer decisions while also enhancing a company's trustworthiness. User reviews have the potential to build trust and motivate others to connect with the business. Businesses benefit from greater customer engagement in the long run and hence to find out the positive and negative reviews is important for business, so this is second dataset we are taking as it serves a real-world problem. The dataset is about coffee shop reviews that is collected from Yelp.com. The dataset contains the coffee shop name, the actual content of review is in column of review text followed by the rating given by customer under 5.0. It is a large dataset containing of 7621 rows, beneficial for training classifiers.

3.1.3. Hate Tweet Dataset

The rise of social media platforms like Twitter has led to an increase in cyberbullying, which is growing more common. Recently, Twitter bullying has received a lot of attention as a possible contributor to an increasing number of suicides [51]. Violence that is spread repeatedly and over time by a group or individual using electronic forms of communication is classified as "cyberbullying." [52]. This is a real-world problem which can be solved by machine learning by detecting online harassment inside the social media platform Twitter. Many Researchers used machine learning to solve this problem such as Zhao and Mao [53] described how they used an embedding-enhanced bag-of-words technique to detect cyberbullying via participant-vocabulary consistency, other attempts have concentrated on the use of supplementary data to improve text-based hate tweet detection, and according to research [53], incorporating profane terms as features improves machine learning model performance significantly.

3.2. Text Embeddings

Text interpretation and creation should be possible using NLP-powered systems that are able to recognize the words, syntax, and other linguistic features. This is a lot easier said than done, though, as computers can only understand numerical values. Text embeddings, an NLP approach designed to fill the hole, was invented by NLP experts to represent words numerically. Once they've been translated, NLP algorithms may simply digest these learned representations to process textual data. As a result of text embeddings, words are transformed into numerical vectors with actual values. Each word in a sequence (or phrase) is tokenized and transformed into a vector space to do this task effectively. Text embeddings are designed to capture the semantic meaning of words in a text sequence. In this way,

words with comparable numerical representations are given numerical representations that correspond to their meanings.

3.2.1. Word Embeddings

These types of embeddings are those which operates on the frequency of words, means how many times this word has appeared in the document or how many documents are there which are containing that particular word so these are based on statistical models. TF-IDF, Bag of Words, Count Vectorizer comes in statistical model. The two types of word embeddings we are using are:

3.2.1.1 Bag of Words

A Categorical word representation is a simple technique to represent text. Symbolic representations of "1" and "0" are used to represent words. One-hot encoding and Bag-of-words are within the category of categorical word representation (BoW). A bag of words is a graphical representation of the frequency with which words appear in a given text. Aside from word count, we don't pay much attention to grammar or word structure.

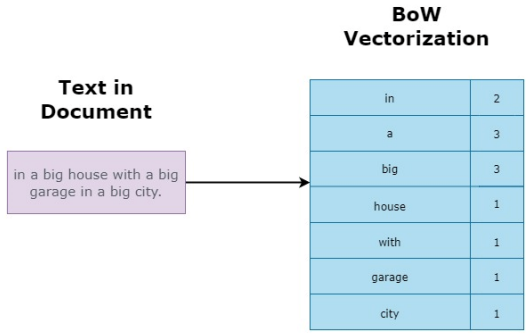


Fig. 2. Bag of Words.

It is referred to as a "bag" of words since all information about the structure or order of words in the text is omitted. The methodology focuses solely on whether recognized terms occur in the document, not on where they appear in relation to each other.

3.2.1.2 TF-IDF

Terms Frequency-Inverse Document Frequency is a Weighted Word representation, TF and TF-IDF are two weighted models based on words' frequency. Term Frequency-Inverse Document Frequency For text representation, the TF-IDF was created to lessen the effect of common terms like "the," "and," and so on across the corpus. Multiplying the frequency of a word's inverse document frequency, as well as the number of times it appears in a document, yields this result. The IDF gives more weight to words that occur more frequently or less frequently. The equation below represents TF-IDF mathematically.

$$TF(\tau,\delta) = \frac{number\ of\ times\ \tau\ appears\ in\ \delta}{total\ number\ of\ terms\ in\ \delta}$$

$$IDF(\tau) = log\ \frac{\nu}{1+\delta f}$$

$$TF\text{-}IDF(\tau,\delta) = TF(\tau,\delta) * IDF(\tau)$$

Where δ represents the document, ν represents the total number of documents, δf refers to the number of documents with the term τ. Because TF-IDF is based on the BOW model, it does not capture the order of words in a document, as well as semantic and syntactical information.

3.2.2. Sentence Embeddings

The semantic meaning of a phrase can be represented using vectors in sentence embedding techniques. This helps the computer understand the text's context, intent, and other characteristics. The two types of sentence embeddings we are using are:

3.2.2.1 Sentence-BERT

Bidirectional Encoder Representations from Transformers (BERT) belongs to a clan of NLP-based language algorithms known as transformers. BERT is a two-variant huge pre-trained profoundly bidirectional encoder-based transformer model. BERT-Base has 110 million parameters, whereas BERT-Large contains 340 million and we use Sentence BERT to encode the text [54].

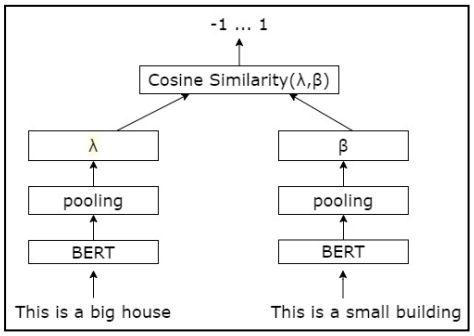


Fig. 3. SBERT.

3.2.2.2 Universal Sentence Encoder

The Universal Sentence Encoder (USE) enables looking up embeddings at the sentence level as simple as looking up embeddings for individual words.

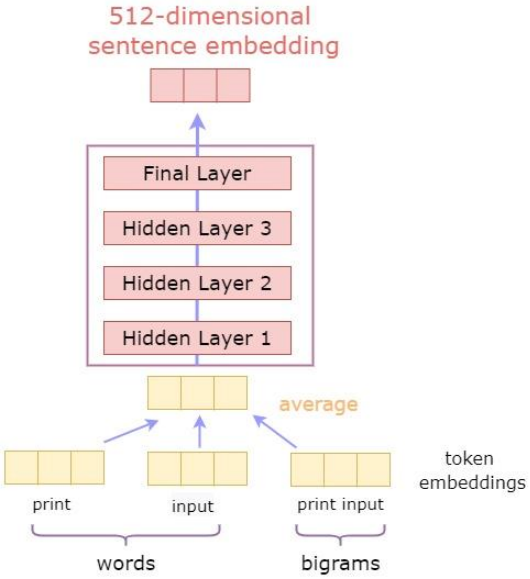


Fig. 4. Universal Sentence Encoder

The Universal Sentence Encoder converts complete sentences or texts into real-number vectors. A Transformer encoder and a Deep Averaging Network (DAN) encoder are available in this product (DAN). Here, we'll be encoding with a DAN encoder (deep averaging network) [55].

3.3. Classifiers

This section covers the classifiers, because we have a supervised task and we have label, and in this research, we are performing a binary classification so we use machine learning algorithms that are statistical in nature so when we get a text we will predict whether it is correct or not, We are using seven different classifiers descripted below.

3.3.1. Linear SVM

Classes would be linearly separable in an ideal situation because the feature space would be splitted into class segments by building a hyperplane that will have the largest margin between the two classes in the training set. The support vectors would be the nearest data points for both classes located parallel to the hyperplane. As a result, Support Vector Machine (SVM) tries to identify the optimal surface to distinguishing between positive and negative training samples. SVM has been employed extensively un hate tweet prediction models, and it has been shown to be successful and efficient so far.

3.3.2. Logistic Regression (LR)

If one or more independent variables may predict an outcome, the statistical approach known as logistic regression can be used to analyse a dataset Using logistic regression, the dependent and independent variables are compared to see which model better captures their connection. In logistic regression, the sigmoid function is used to convert predicted values to probabilities and vice versa. The number is converted to a value between 0 and 1, and then back to its original actual value.

3.3.3. K nearest neighbors (KNN)

KNNs are one of the most basic nonparametric classifiers, although their performance is harmed by nuisance characteristics in high-dimensional settings. basically, the KNN algorithm assumes that similar things exist in near vicinity. Alternatively, the case is allocated to the class with the most members in its K closest neighboring classes, as defined by a distance function, by a majority vote among the case's neighbors. If K is 1, then the case is allocated to the class of the closest neighbor.

3.3.4. Naive Bayes (NB)

NB is a simple Bayesian-based probabilistic classifier. NB makes the strong assumption that instance features are independent of one another, but it produces results that are equivalent to much more complex classifierswhich is why it is frequently used as a baseline for many machine learning tasks. Furthermore, the independence assumption simplifies the training process by reducing it to the model learning the attributes separately, which greatly reduces the temporal complexity of huge datasets.

3.3.5. Decision Tree (DT)

DT is an easy yet efficient supervised learning algorithm in which data points are constantly split based on certain characteristics and/or the problem that the algorithm is looking to solve. The root node of a decision tree is always at the top of the structure, whereas the outcomes are depicted by the tree leaves. Using the decision tree technique, we

begin at the root of the tree and divide the data based on the characteristic that delivers the most information gain. Afterwards, we may repeat this process for each of the child nodes until the leaves are completely pure. These results show that each leaf node has samples of the same type.

3.3.6. Random Forest (RF)

RF is a classification algorithm that uses the average of a number of decision trees on different subsets of a dataset to enhance the dataset's prediction accuracy. basically, getting dependent on a single decision tree, the random forest gathers predictions from each tree and predicts the eventual output on the basis of majority votes of predictions. The more trees in the forest, the higher the accuracy and the lower the risk of overfitting.

3.3.7. Stochastic Gradient Descent (SGD)

In SGD learning linear classifiers using convex loss functions, such as SVM and Logistic regression. Because the coefficients are updated during training rather than at the conclusion of training, it has been effectively used to big datasets. There are several loss functions and penalties that may be used to penalize classification in the SGD classifier.

## IV. PROPOSED METHODOLOGY

Before understanding the proposed methodology, we need to understand why text data augmentation is necessary. Basically, in lot of cases we have low volume dataset, and the quantity and diversity of data are important factors in the effectiveness of most machine learning models. It helps machine learning models perform better and lead to better outcomes. The data augmentation techniques enrich and supplement the data, allowing the model to perform better and more precisely. In many cases we don't have enough data to build reliable models, thus we end up with data that has a persistent class imbalance.

Suppose we have few types of sentences in dataset and if a different type of sentence which is not occurred much in the dataset comes up for prediction, then that sentence might not be treated properly, and might not get a correct prediction. So, if the dataset is vast then it’s likely to cover a lot of different types of sentences and when a new different sentence comes up for predication, then that similar type of sentence might be covered earlier in the current dataset, and the probability of getting the correct prediction for that new sentence is also increases, so that is why increasing the size of dataset, more precisely increasing a minority class is most important. So, in case we have low size dataset then data augmentation techniques come very handy. Because the dataset we are considering is text dataset, therefore we are using text data augmentation and the classification task we are doing is on text datasets.

Now augmentation is basically distorting the sentences a little such that it becomes a new sentence from the original sentence and then adding that new set of sentences to the original dataset, thus the size also increases, and we have more types of sentences, we do not directly clone the original sentence, we change a little and then create a new set of sentences and add that to increase the size of data. The resultant augmented data distribution should be neither too similar nor too dissimilar to the original. Therefore, we suggest the ensemble of data augmentation as an effective DA technique that aims for a balance. We compare our suggested strategy to a set of four simple text augmentation techniques for further classification purpose on three datasets.

Novelty –

The Novelty of the work is that which augmentation technique provides best results for the certain dataset, in this work we have taken a real-world problem of hate tweet detection, email spam detection and Customer reviews and for these tasks we have very small size data which give poor results and to mitigate this problem we augment this data by the best augmentation technique and increase the size and get higher classification accuracy.

We have the text datasets, and we are distorting/modifying the dataset using four different kinds of augmentation techniques and then adding that augmented dataset to the original dataset to increase the volume of dataset and later using that for classification purposes. So basically, we use that on a task then we and compare the accuracies to check that augmentation has increased the performance of the classifiers on the dedicated classification task.

We have performed the work in four phases too carry out different tasks, we have described each phase in detail.

*4.1. Methodology Phases*

The Important phases of the task carried out are:

Phase 1 – Data Pre-Processing

Phase 2 – Analysis using original data.

Phase 3 – Analysis with Random augmentation technique

Phase 4 – Analysis of Ensemble Data Augmentation

Let's look at each phase one by one:

4.1.1. Data Pre-Processing

To clean the datasets, Pre-processing is done using regular expressions in python. Figure 5 shows Algorithm for pre-processing of tweet dataset:

Step-1: Loading the raw tweets / mails / reviews: Firstly, we will import the regular expressions that will perform pre-processing, and we will load the dataset for further preprocessing.

Step 2: Removal of '@' symbol and URLs: The URLs and words that begin with the '@' symbol are removed in the second step. The entire word might be omitted because the '@' symbol is always followed by a username, and other words starting with '@'which is meaningless.

Step 3: Removal of Unicode characters: The Special Characters such as "Pyéthonò!" are removed to"python" because only English text is considered.

Step 4: Removal of Hashtags (#): Hashtag is a term that starts with '#' and lends a subject or tag to a tweet /mail /review. This word may provide some information, but it is not particularly crucial. As a result, the entire term beginning with a # sign is removed, and the tag is treated like any other word in the text.

Step 5: Lowercase: All the words in the given text are rewritten in lowercase letters.

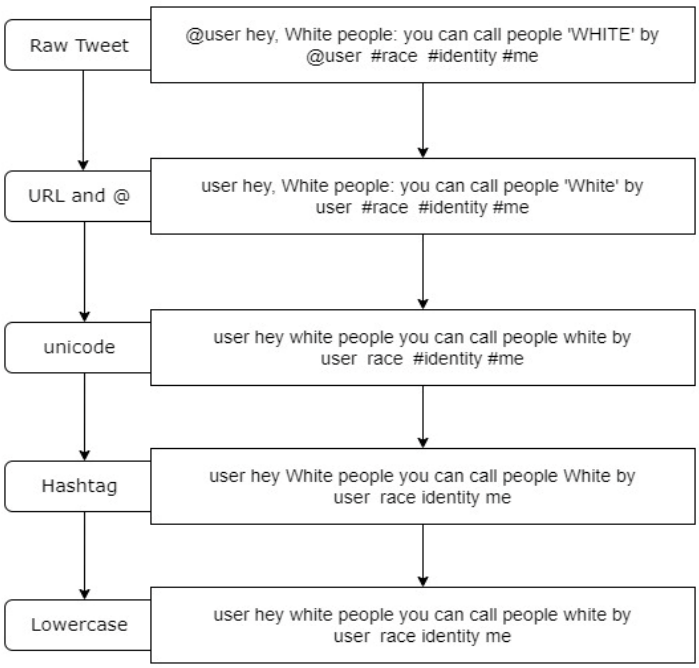


Fig. 5. Preprocessing Steps.

The Preprocessing of Spam Email dataset will also be same as the preprocessing of hate tweet data, but while preprocessing we will change the labels of the dataset from Spam and Non-Spam to 0 and 1 where 0 represents non-Spam mail and 1 represents spam mail.

Similarly, the preprocessing of Yelp Coffee Reviews dataset will too be same as the preprocessing of hate tweet data, but in this dataset, there will be few changes, In this dataset we will first perform data cleaning where we only select the review text and rating given. Now in this dataset

the ratings are in multi class, so we will convert those into binary classes, we will consider ratings 0 to 2 as poor and hence we will mark as 1 and similarly ratings 4 and 5 will be considered as great hence we will mark as 0. All the ratings labelled as 3 will be discarded to create a proper class imbalance for this experimentation and then we will change review_text to review and rating column to label.

4.1.2. Phase 2 – Analysis using original data

In this phase we first test the dataset performance without any augmentation technique applied on it. We will be only using the original datasets of hate tweet as well as spam mails and Yelp Coffee reviews for classification purposes. The datasets will be split into training and validation sets. And thereafter we will use four text embedding techniques separately to covert text into numerical arrays, by tokenizing and encoding and transforming every word in a series into a vector space. so, compare and understand the semantic meaning of a word in a text sequence.

The four text embedding techniques we are using are, Sentence BERT (SBERT), Universal Sentence Encoder (USE), Bag of Words (BoW), and TF-IDF (Term Frequency-Inverse Document Frequency). After applying text embedding, we will use seven different supervised machine learning algorithms that are statistical in nature so when we get a new text, we will predict whether it is positive or negative in all dataset terms because all datasets serves the real-world problem. Seven different classifiers that are used for classification purpose are: Support Vector Classifier (SVC), K Nearest Neighbor (KNN), Naive Bayes Classifier (NB), Logistic Regression (LR), Random Forest (RF), Decision Tree (DT), Stochastic Gradient Descent (SGD). Each classification algorithm will then provide respective evaluating results. The architectural diagram of analysis of original data is given below in figure 6.

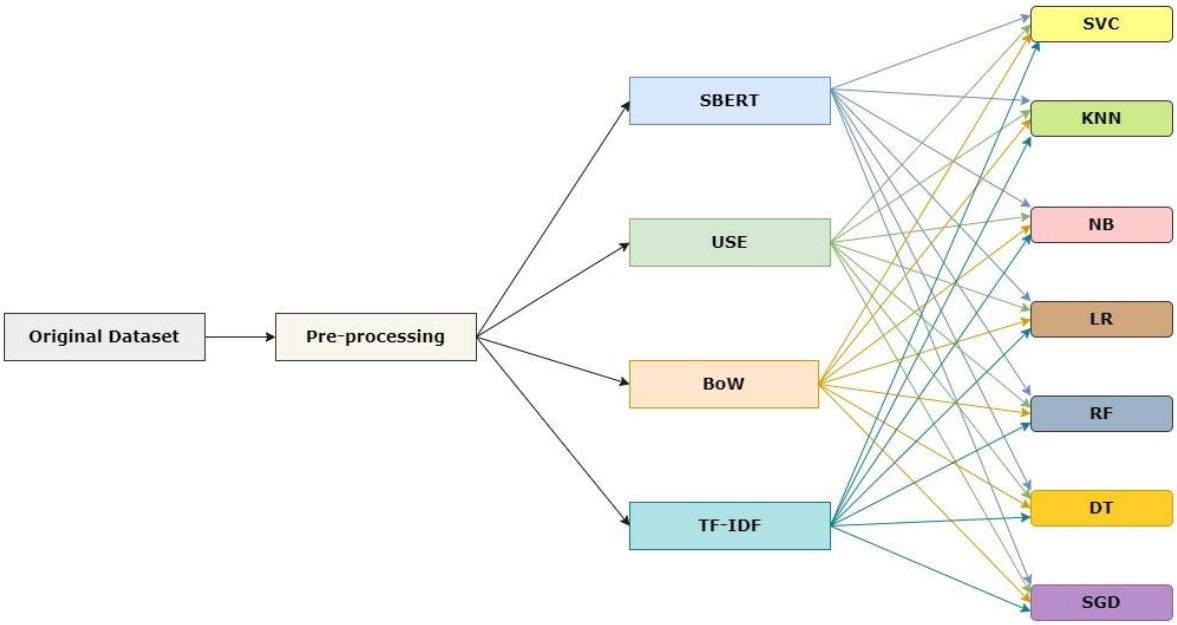


Fig. 6. Flow of Analysis with original data.

4.1.3. Phase 3 – Analysis with Random augmentation technique

In this phase we will test the datasets performance with any random augmentation technique applied on it. We will be using the Keyboard Error Injection augmentation technique to augment the original datasets of hate tweet, spam email and Yelp Reviews.

Class Imbalance –

To mitigate the class imbalance problem, in this research we are only applying augmentation to a minority class only then later merge the augmented dataset to original dataset to have new enhanced augmented dataset for classification purpose.

After improving class imbalance, the dataset will be split into training and validation sets. And

thereafter we will again use four text embedding techniques separately to covert text into numerical arrays, by tokenizing and encoding and transforming every word in a series into a vector space. So, compare and understand the semantic meaning of a word in a text sequence. The four text embedding techniques we are using are, SBERT, USE, BoW, and TF-IDF. After applying text embedding, we will use seven different supervised machine learning algorithms that are statistical in nature so when we get a new text, we will predict whether it is 0 class or 1 class respective to datasets. Seven different classifiers that are used for classification purpose are – SVC, KNN, NB, LR, RF, DT and SGD. Each classification algorithm will then provide respective evaluating results. The architectural diagram of analysis with random augmentation technique is given below in figure 7.

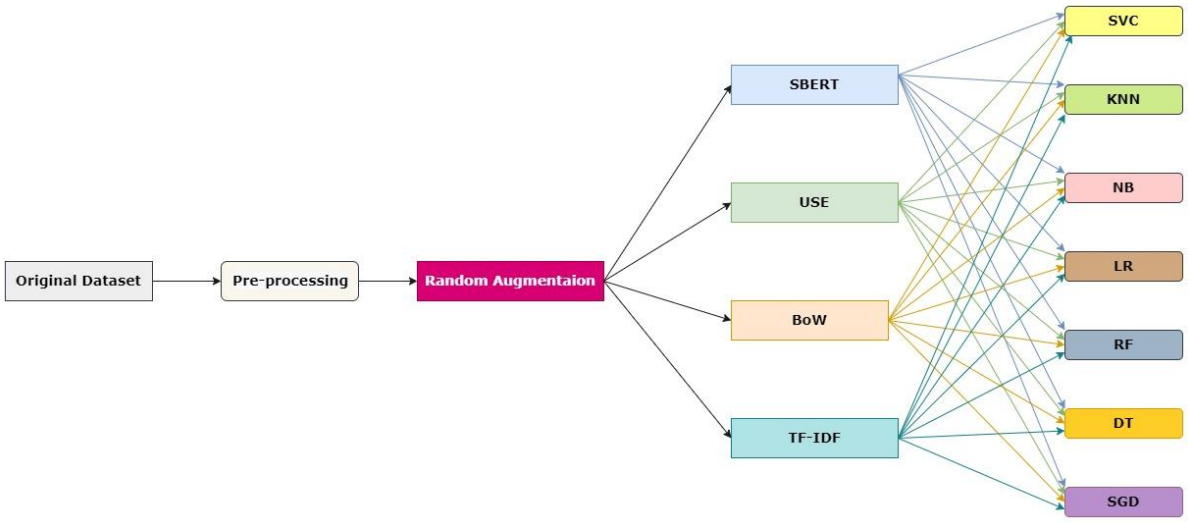


Fig. 7. Flow of Analysis with Random Augmentation Technique

4.1.4. Phase 4 – Analysis of Ensemble Data Augmentation

This is the major phase carried out in our research experiment. In this phase we will test the datasets performance by applying ensembles of data augmentation techniques applied on it. We will be using four different augmentation techniques for this purpose. Random Deletion, Random Swap, Synonym Substitution and Antonym Substitution are the four augmentation techniques are used to augment the minority class of original dataset of all three datasets.

After applying all four augmentation technique, we will encode all the augmented results separately using SBERT, thereafter using cosine scores of original dataset embeddings and of all other augmented results embeddings we will compare the cosine similarity of all augmented results to original dataset, then whichever augmentation technique will have the highest cosine similarity will be passed on to merge with original dataset to have new enhanced efficient augmented dataset for classification purpose. Then dataset will be split into training and validation sets. And thereafter we will use four text embedding techniques separately to covert text into numerical arrays, by tokenizing and encoding and transforming every word in a series into a vector space to compare and understand the semantic meaning of a word in a text sequence and also to find out what effect does different text embeddings do on enhanced dataset.

The four text embedding techniques we are using are, SBERT, USE, BoW, and TF-IDF. After applying text embedding, we will use seven different supervised machine learning algorithms that are statistical in nature so when we get a new text, we will predict whether it is class 1 or class 0 depending on the dataset. Seven different classifiers that are used for classification purpose are: SVC, KNN, NB, LR, RF, DT and SGD. Each classification algorithm will then provide respective evaluation results. The architectural diagram of analysis of original data is given below in figure 8.

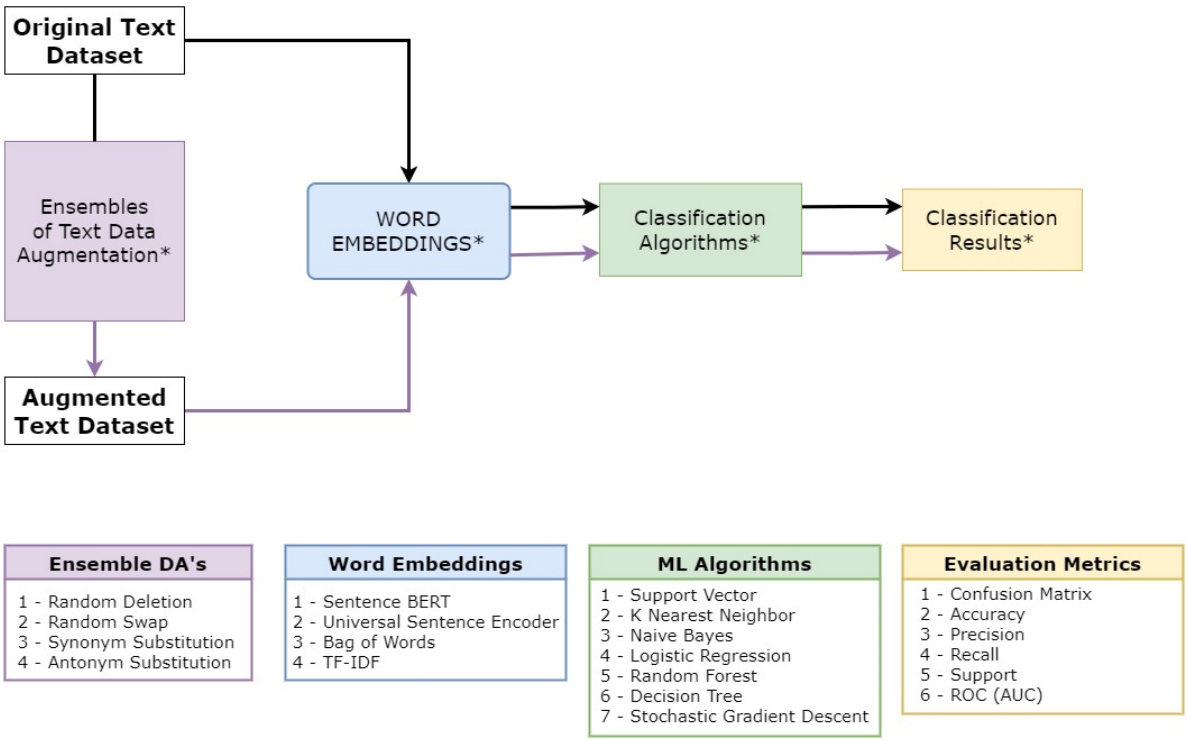


Fig. 8. Flow of Ensemble Data Augmentation.

4.2. Proposed Algorithm

Below given is the stepwise algorithm for the tasks carried out.

Step 1: Considering the datasets for experimentation.

Step 2: The considered data are in raw form. To perform and classification procedure, at first the text is needed to go through pre-processing. Removal of URLs, hashtags, Unicode characters are all processes in the pre-processing process.

- Importing all the essential dependencies.
- Create a data frame for tweets and persevering it in memory.
- Removal the junk characters (#, urls, @) and convert the whole data frame to lowercase.
- Creating new column of named clean_text for pre-processed text in the data frame.

Step 3: Classification using Original Dataset

- Importing all the desired dependencies.
- Splitting the dataset into a training set and testing set.
- Applying Text embedding techniques
- Tokenizing and encoding word in a series into a vector space.
- Applying classification algorithms for the training dataset
- Predict the outcome of the tested dataset.
- Predict the outcome and print the result.

Step 4: Analysis using Randomly Augmented Dataset

- Importing all the desired dependencies.
- Applying Keyboard Error Injection augmentation
- Concatenating original dataset and augmented dataset (minority class)
- Split the enhanced dataset into a training set and testing set.
- Applying Text embedding techniques
- Tokenizing and encoding word in a series into a vector space.
- Applying classification algorithms for the training dataset
- Predict the outcome of the tested data set.
- Predict the outcome and print the result.

Step 5: Analysis using Ensembling of Data Augmentation Techniques

- Importing all the desired dependencies.
- Applying ensembles of data augmentation
- Encoding all the augmented results (minority class)
- Compare cosine scores of augmented sets and original dataset.

- Original dataset with highest cosine similarity augmented dataset
- Splitting the efficient enhanced dataset into a training set and testing set.
- Applying Text embedding techniques
- Tokenizing and encoding word in a series into a vector space.
- Applying classification algorithms for the training dataset
- Predict the outcome of the tested data set.
- Predict the outcome and print the result.

4.3. Proposed Flowchart

Below is the proposed flowchart: -

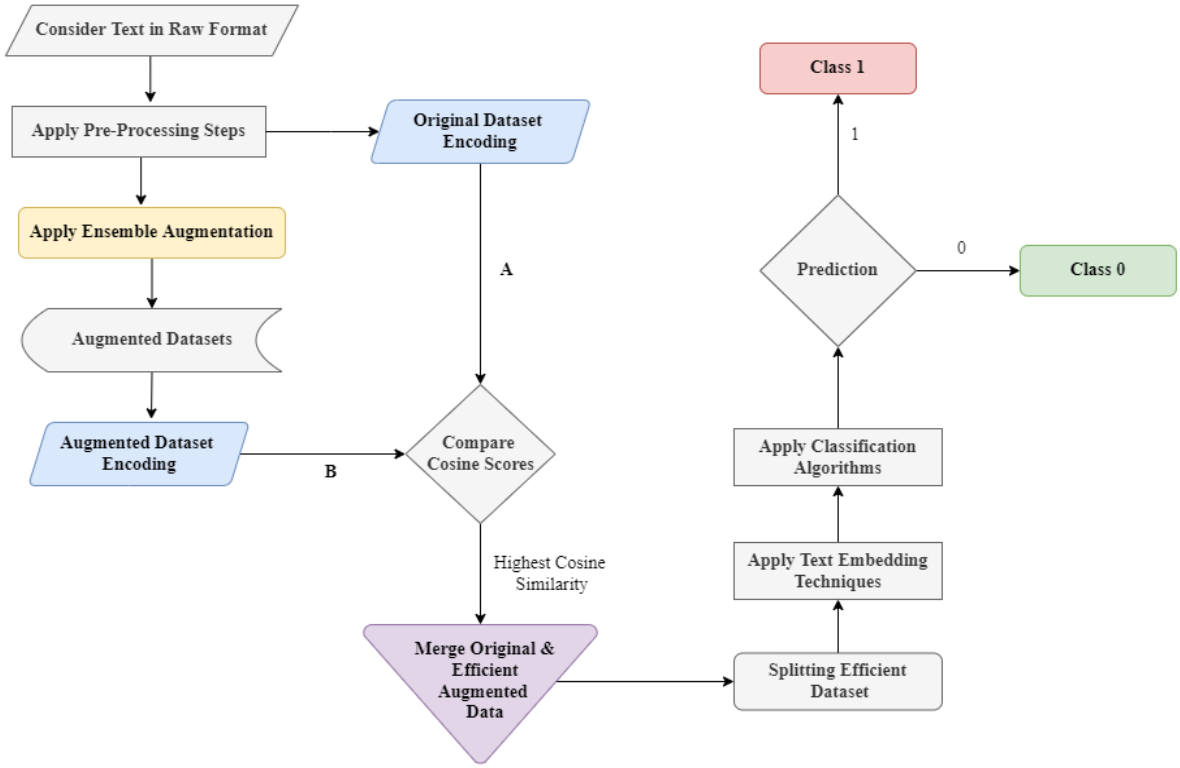


Fig. 9. Proposed Flowchart

## V. IMPLEMENTATION, RESULTS & DISCUSSION

We have performed experimentations on three datasets by applying data augmentation and later classifying for prediction. After applying preprocessing steps, we will use nlpaug library in python to import various data augmentations and using those only on minority classes of original datasets to improve class imbalance.

Ensembling four DA techniques, Random Deletion, Random Swap, Synonym Replacement and Antonym Replacement on all three datasets, we got four augmented results. Now to automatically select the efficient result we have used cosine similarity metric.

Cosine Similarity –

In NLP, cosine similarity is perhaps one of the metrics used to compare the text similarity of two texts, regardless of their size. A vector representation of a text is created and in n-dimensional vector space, the text later represented. The cosine angle θ of among two n-dimensional vectors is measured by the Cosine similarity metric. The range of cosine similarity will range between 0 and 1. If two vectors said to be similar, the Cosine similarity will be 1 else 0. In Mathematical terms the vectors of two vectors is given as:

$$\text{Similarity}(A,B) = \cos(\theta) = \frac{A.B}{\|A\| \, \|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Now, after encoding original data and encoding all four augmented data then applying cosine similarity metric, the results that came out on all three datasets are given in Table 4 below:

TABLE IV. Cosine Similarity w.r.t original embeddings

| Augmentation | Similarity (Spam Email) | Similarity (Yelp Coffee Reviews) | Similarity (Hate Tweet) |
|---|---|---|---|
| Random Delete | 85.17 % | 94.58 % | 81.83 % |
| Random Swap | 97.71 % | 98.70 % | 62.19 % |
| Synonym Replacement | 84.35 % | 92.72 % | 85.76 % |
| Antonym Replacement | 75.68 % | 93.38 % | 80.51 % |

From the table is visible Random Swap Augmentation technique has the highest cosine similarity w.r.t original encoded sentences for Spam email i.e., 97.71% and for Yelp Coffee reviews 98.70% whereas Synonym Replacement Augmentation technique has the highest cosine similarity for Hate tweet dataset which means the efficient augmentation for Spam Email Dataset and Yelp Reviews Dataset is Random Swap and efficient algorithm for Hate Tweet Dataset is Synonym Replacement. So now we took the efficient augmentation of minority class and concatenated with original data respectively and therefore we improved class imbalance problem. The comparative plot is shown in figure 10.

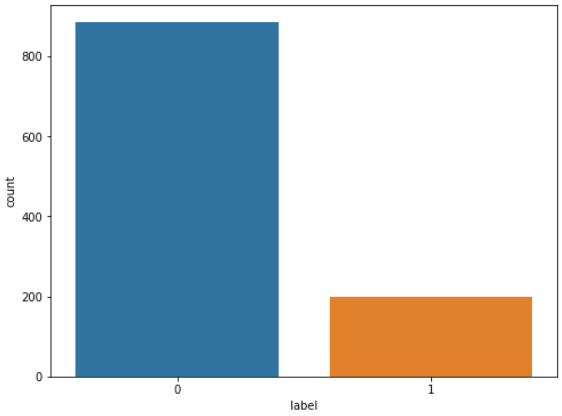


Fig 10 (a). Before Data Augmentation

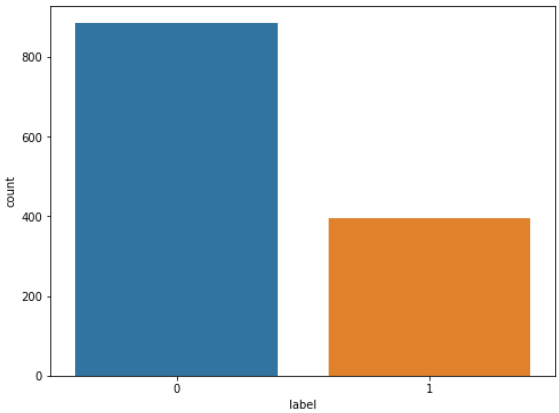


Fig 10 (b). After Data Augmentation

Below we present the plotted word cloud of both classes from Hate Tweet dataset.

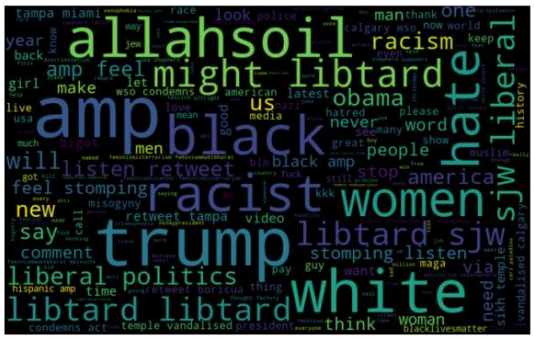


Fig 11 (a). Hate Tweet Word Cloud

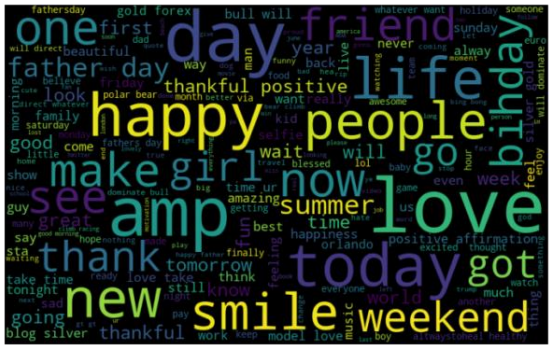


Fig 11 (b). Non-hate Tweet Word Cloud

Taking out the ensemble of data augmentation, the efficient data with highest similarity is shown in below figure 12.

| tweet | clean_text | effcient_data |
|---|---|---|
| @user #cnn calls #michigan middle school 'buil... | cnn calls michigan middle school 'build the... | cnn calls michigan middle school build the ' '... |
| no comment! in #australia #opkillingbay #se... | no comment in australia opkillingbay se... | comment no australia in opkillingbay helpcoved... |
| retweet if you agree! | retweet if you agree | retweet agree if you |
| @user @user lumpy says i am a . prove it lumpy. | lumpy says i am a prove it lumpy | says i lumpy am a it prove lumpy |
| it's unbelievable that in the 21st century we'... | it's unbelievable that in the st century we'... | it ' s unbelievable that in the century st ' w... |
| ... | ... | ... |
| lady banned from kentucky mall. @user #jcpenn... | lady banned from kentucky mall jcpenny ke... | Lady banned from mall kentucky jcpenny kentucky |
| @user omfg i'm offended! i'm a mailbox and i'... | omfg i'm offended i'm a mailbox and i'm pro... | omfg ' i m offended ' i a m and i mailbox m ' ... |
| @user @user you don't have the balls to hashta... | you don't have the balls to hashtag me as a ... | don you ' have t the balls to hashtag me a as ... |
| makes you ask yourself, who am i? then am i a... | makes you ask yourself who am i then am i a... | you makes ask yourself am who i am then anybod... |
| @user #sikh #temple vandalised in in #calgary,... | sikh temple vandalised in in calgary wso... | sikh vandalised temple in in calgary wso act c... |

Fig 12. Efficient Data (Hate Tweet Dataset)

After the data was efficiently augmented then that data was passed through four text embeddings to compare which embedding perform better with augmented data for machine learning classification.

5.1. Evalation Metrics

An evaluation of the models is essential when the classifiers are being built. The purpose is to have a better understanding of the classifier's performance on a global accuracy, which will mask the faults in one class of a multiclass problem. The classification report is used to assess all the classification models used across the report and select the ones with the best classification metrics or the most balanced. True and false positives, as well as true and false negatives, will be utilized to calculate metrics. When both the actual and estimated classes are positive, a genuine positive occurs, whereas a false positive occurs when the actual class is negative, but the estimated class is positive. According to [56], the following is the evaluation:

5.1.1. Confusion Matrix (CM)

ML classification jobs with two or more output classes may be evaluated. Expected and actual outcomes are combined in this table. A classification model's performance on a set of test data for which the real values are known is typically described using a CM.

5.1.2. Accuracy

A classifier's accuracy is simply the number of times it correctly predicts the outcome of a given experiment. The accuracy of a forecast is calculated as the number of right predictions divided by the total number of forecasts.

5.1.3. Prediction

Prediction accuracy shows how many of the situations that were accurately predicted turned out to be right. It helps to have better precision when it comes to detecting False Positives rather than False Negatives.

5.1.4. Recall

Recall is the percentage of positive occurrences predicted by our model that really occurred. When False Negative is more risky than False Positive, it's an excellent statistic to utilise.

5.1.5. F1 score

When attempting to find a balance between Precision and Recall, as well as when there is an unequal class distribution, F1 score is required. As a result, the F1score is a weighted average of the two criteria, with 1.0 being the best and 0.0 being the worst.

5.1.6. Support

It is a measure of how many actual instances of a certain class there are in the dataset. Rather of focusing on model differences, it examines how performance is evaluated.

5.1.7. ROC

Receiver operator characteristic (ROC) is a probability curve that compares the TPR (True Positive Rate) against the FPR (False Positive

Rate) at different threshold levels and distinguishes the "signal" from the "noise. "

5.2. Experimental Results

The results came are very interesting to see. Below we give the results came out after classifier trained with efficient augmentation and best encoded technique.

5.2.1. Hate Tweet Dataset

The Hate Tweet was efficient augmented by Synonym Replacement Augmentation technique and upon classifying with efficient augmented data, RF algorithm got the highest accuracy of 96.73% with TF-IDF embedding method, this is 1.24% more than original dataset which also classified labels correctly also there was so there has been increase in accuracy in almost every algorithm.

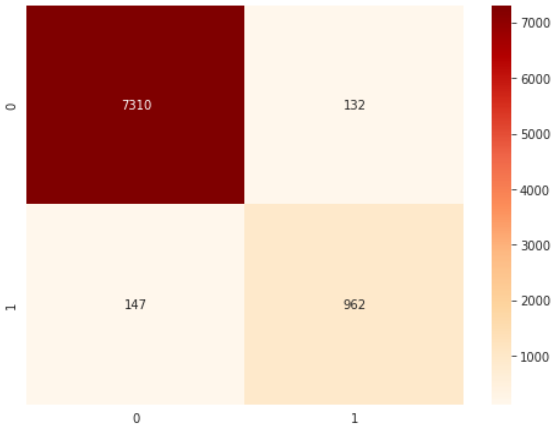


Fig 13. RF Confusion Matrix Hate Tweet Dataset

Here the 7310 tweets were correctly classified as Positive tweets and 962 emails were correctly classified as Negative tweets during testing. The AUC score for this model improved to 97.21% and the ROC Curve is shown in below figure.

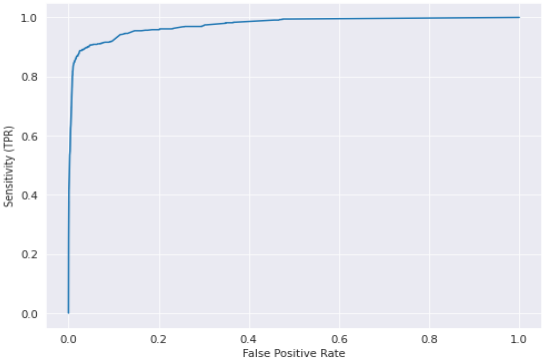


Fig 14. ROC Curve RF Hate Tweet Dataset

5.2.2. Spam Email Dataset

The Spam Email was efficient augmented by Random Swap Augmentation technique and upon classifying with efficient augmented data, SGD algorithm got the highest accuracy of 99.06% with USE embedding method, this is 0.8% more than original dataset which also classified labels correctly also there was so there has been increase in accuracy in almost every algorithm.

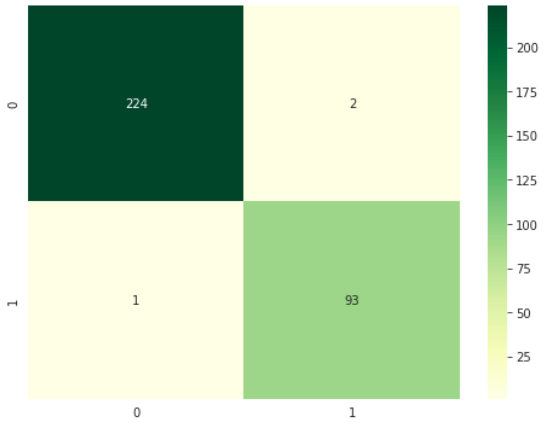


Fig 15. SGD Confusion Matrix Spam Email Dataset

Here the 224 emails were correctly classified as non-spam and 93 emails were correctly classified as Spam during testing. The AUC score for this model improved to 99.01% and the ROC Curve is shown in below figure.

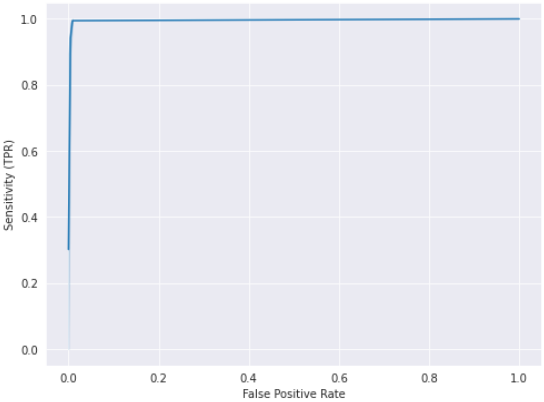


Fig 16. ROC Curve SGD Spam Email Dataset

5.2.3. Yelp Coffee Reviews Dataset

The Yelp Coffee Reviews Dataset was efficient augmented by Random Swap Augmentation technique and upon classifying with efficient augmented data, RF algorithm got the highest

accuracy of 97.21% with TF-IDF embedding method, this is 3.55% more than original dataset which also classified labels correctly also there was so there has been increase in accuracy in almost every algorithm.

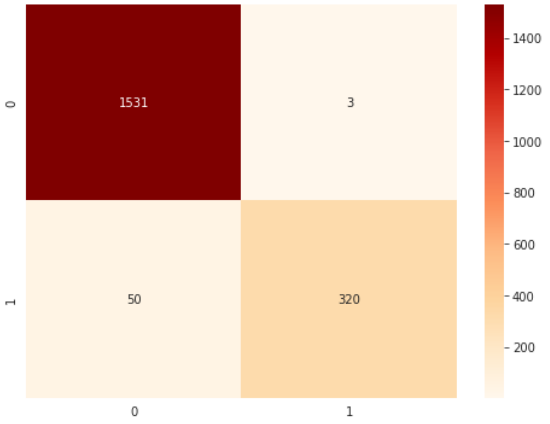


Fig 17. RF Confusion Matrix Reviews Dataset

Here the 1531 reviews were correctly classified as Positive Reviews and 320reviews were correctly classified as Negative Reviews during testing. The AUC score for this model improved to 98.12% and the ROC Curve is shown in below figure.

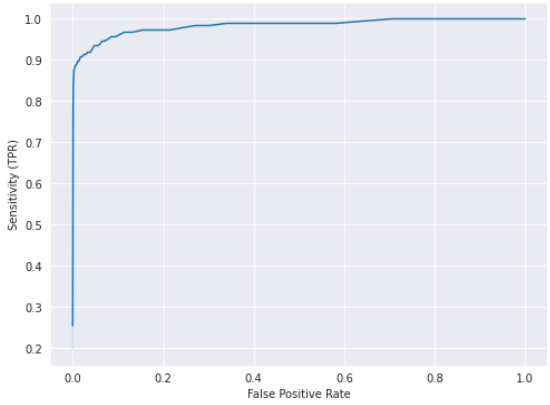


Figure 18. RF Confusion Matrix Reviews Dataset

5.3. Performance Analysis

We have applied the machine learning algorithms on original dataset, on random augmented dataset and on efficiently augmented dataset on all three datasets. We have results of all the phases now we compare the performance of classifiers and embedding techniques w.r.t to augmentation techniques on all three datasets in a visualized manner, we have found interesting insights from this performance analysis.

5.3.1. Spam Email Dataset Analysis

On Spam Email Dataset the below figures explain the performance of augmentation techniques with respect to Text embedding methods and Classifiers.

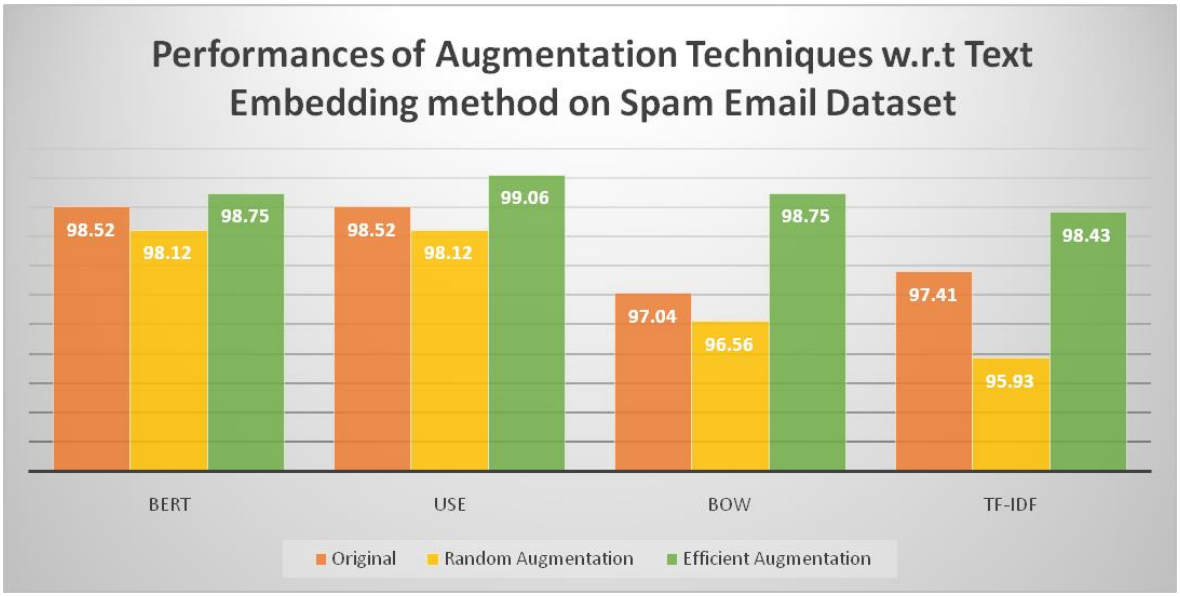


Fig 19. Performance Analysis of Embedding techniques on Spam Email Dataset

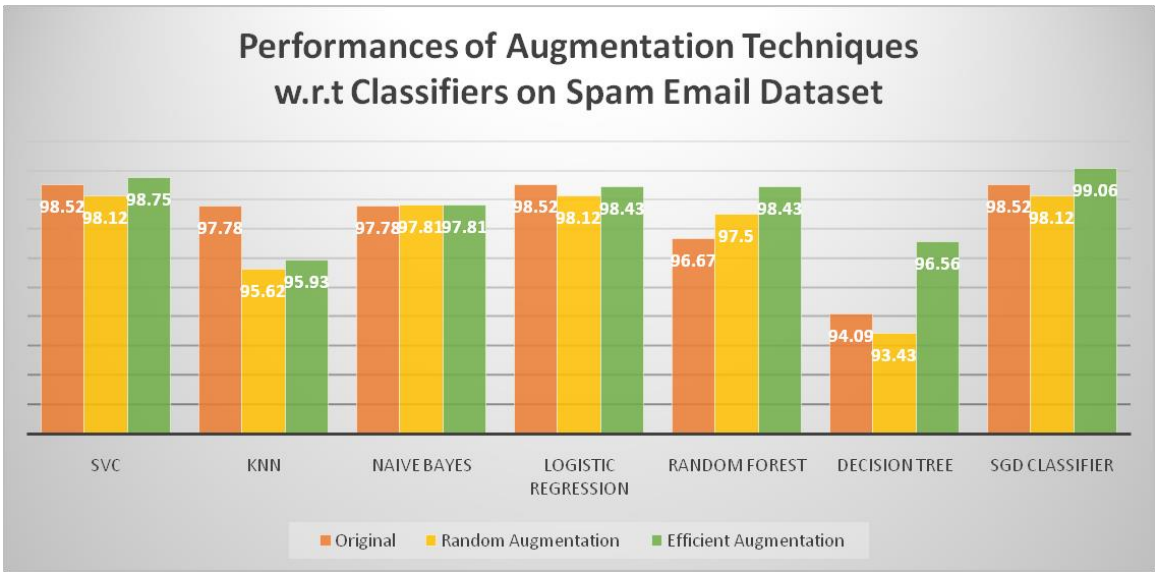


Fig 20. Performance Analysis of Classifiers on Spam Email Dataset

From the Figure 19 and Figure 20 it is clearly visible that in this dataset even though the data is expanded using a random augmentation the accuracies are decreasing, this is because the augmentation technique we used as random augmentation i.e., Keyboard Error Injection comes under non-linguistic category which means it does not maintain semantics of text, that's why we can see that efficient augmentation improves the accuracies in almost every algorithm. From Figure 19 we can also see that SBERT and USE embedding techniques are performing much better in comparison to BoW and TF-IDF.

5.3.2. Yelp Reviews Dataset Analysis

On Yelp Reviews Dataset the below figures explain the performance of augmentation techniques w.r.t Text embedding methods and Classifiers.

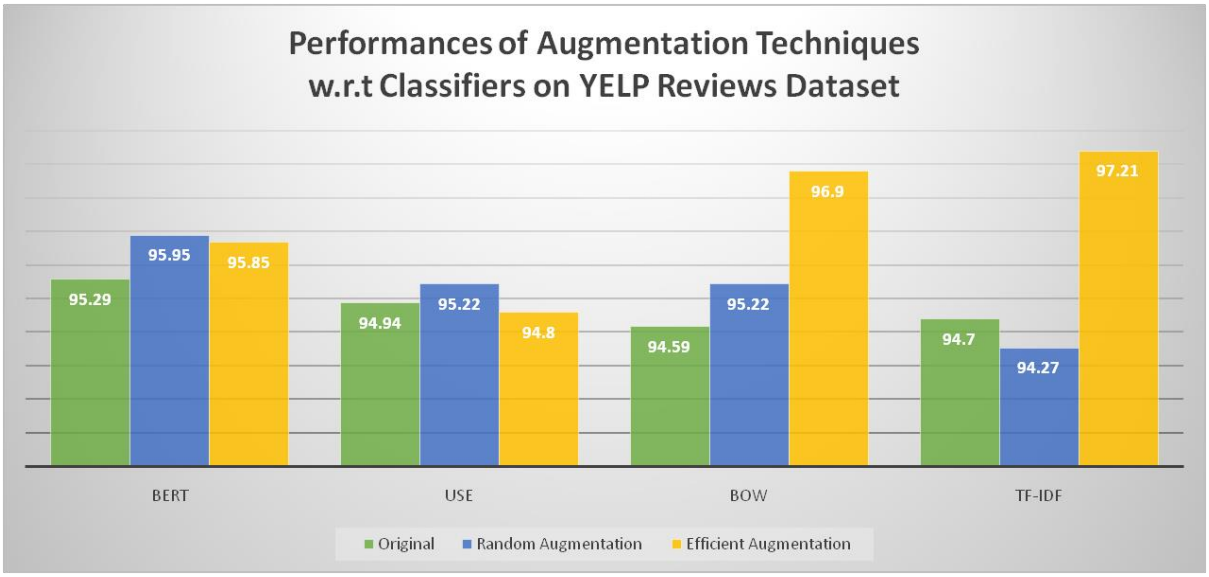


Fig 21. Performance Analysis of Embedding Techniques on Yelp Reviews Dataset

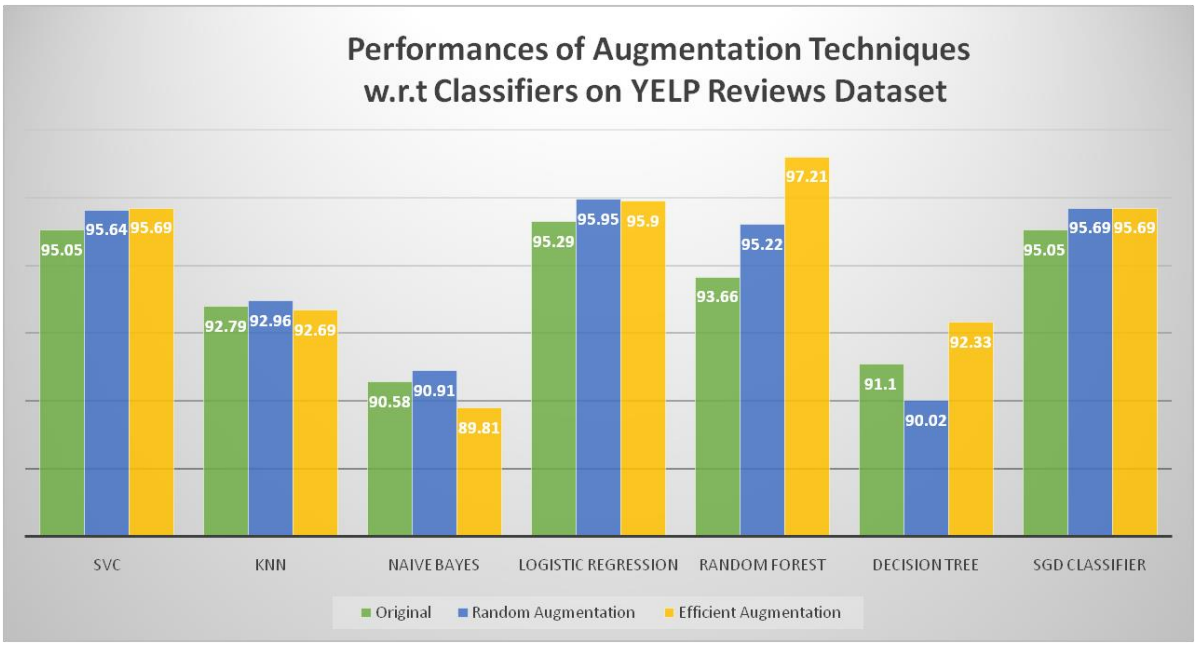


Fig 22. Performance Analysis of Classifiers on Yelp Reviews Dataset

From the Figure 21 and Figure 22 here is an interesting result it is clearly visible that in this dataset the data expanded using a random augmentation, the accuracies are comparatively equal to efficient augmentation. From Figure 21 we can also see that here BoW and TF-IDF embeddings techniques have performed exceptionally well, SBERT and USE embedding techniques did not performed well and this shows that the text embedding techniques also play a great role and is directly proportional in improving classification accuracy. From Figure 22 we also can see that DT Classifier and NB Classifier have performed badly on this dataset in all phases while RF has performed exceptionally well on this dataset.

5.3.3. Hate Tweet Dataset Analysis

On Hate Tweet Dataset the below figures explain the performance of augmentation techniques w.r.t Text embedding methods and Classifiers.

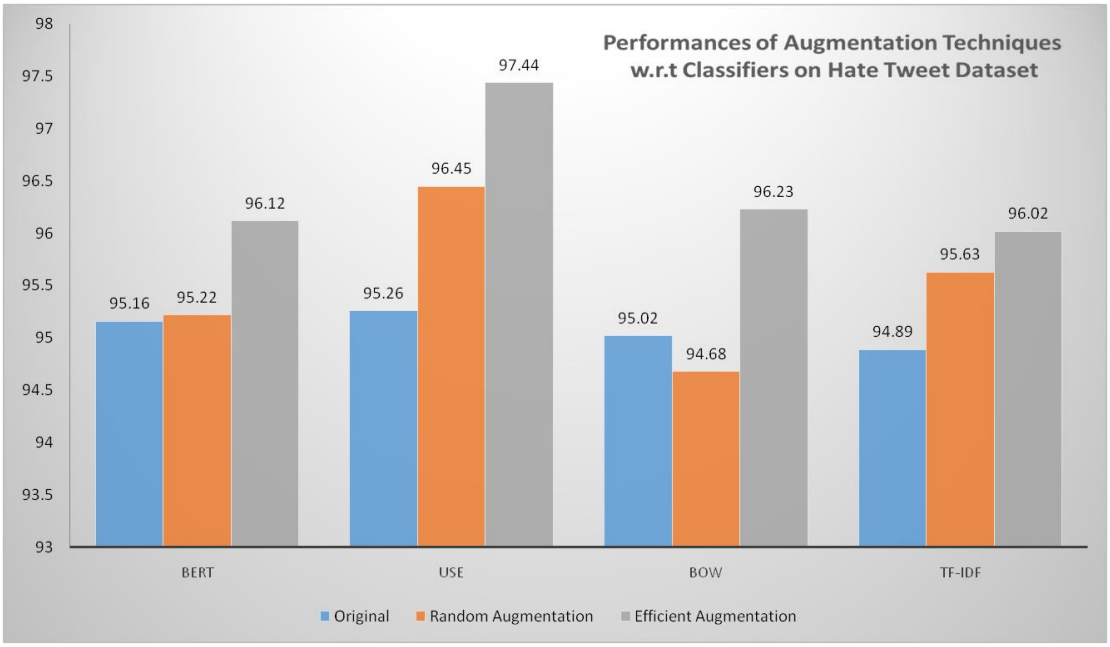


Fig 23. Performance Analysis of Embedding Techniques on Hate Tweet Dataset

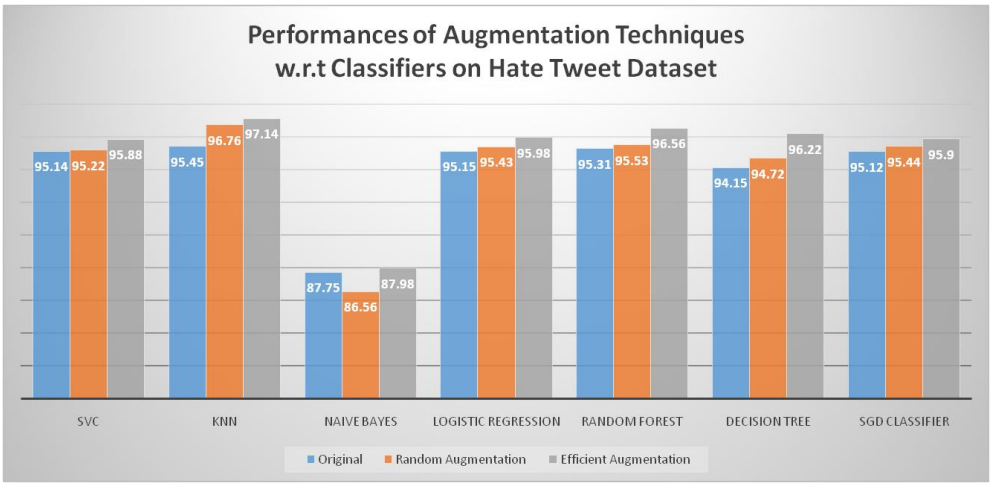


Fig 24. Performance Analysis of Classifiers on Hate Tweet Dataset.

From the Figure 23 and Figure 24, it is clearly visible that in this dataset, the original dataset has the least accuracy followed by the random augmentation and the best accuracies are achieved through efficient augmentation and that's why we can see that efficient augmentation improves the accuracies in almost every algorithm. From Figure 23 we can also see that USE embeddings technique has performing a lot better in comparison to other embedding techniques. From Figure 24 we can see that KNN has performed well in this dataset and similarly all other algorithms are performing better with efficient augmentation apart from NB classifier. NB Classifier has performed too badly on this dataset.

5.4. Discussion

In this research we have augmented only single minority class in a dataset to improve the class imbalance, Ensemble Data Augmentation can also applied to whole original dataset to increase the size of dataset. From the results it is absolute clear that the only applying data augmentation to text does not guarantee for improved classification results, there is a possibility that the augmentation can be of nonlinguistic category and can change the actual meaning. Hence text embedding techniques also plays a great role after data augmentation to encode the text for classification. In this research SBERT and USE embedding techniques gave a constant improvement in classifier accuracy in using original and augmented data, also a TF-IDF has improved the classification accuracy only with augmented data. BoW performed lowest in encoding the text. All the classifiers used have shown an improved classification accuracy except Naïve Bayes, NB has performed badly while classifying with augmented data as shown in Fig 25. NB classifier is wrongly classifying when error-based augmentation techniques is applied. Ensemble Data Augmentation has improved the accuracy around 1-3% on average on all three datasets.

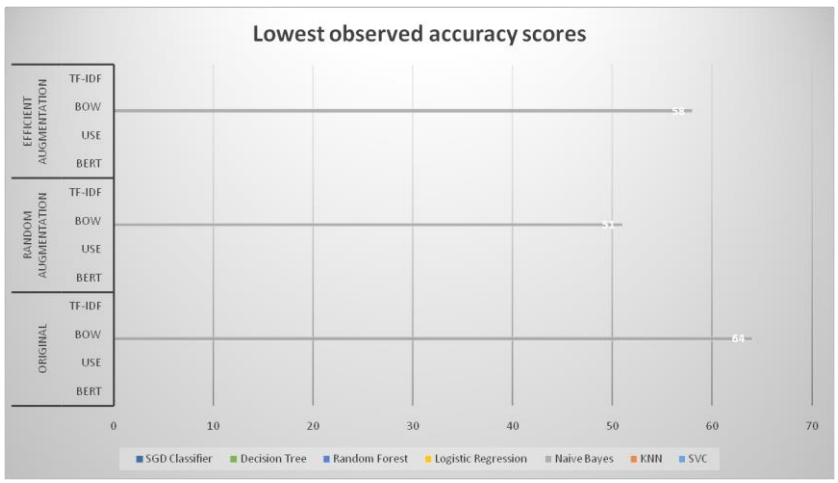


Fig 25. Poor Performance of Naïve Bayes Classifier.

## VI. CONCLUSION AND FUTURE WORK

During this research, various experiments were carried out for three different datasets. Seven Different classifiers were used in this research (Linear SVC, KNN, NB, LR, RF, DT and SGD). Four different text embedding techniques were employed to encode text to numeric form. The results show that ensemble of data augmentation has improved the accuracy with baseline and with any other random augmentation technique. Four augmentation technique that were used during ensembling, out of them for Spam Email and Yelp reviews Random Swapping comes out to be efficient technique with cosine similarity of 97.17% and 98.70 % respectively, but for Hate Tweet Dataset Synonym Substitution comes out to be efficient with similarity of 85.76%.

Upon supervised classification by balancing the minority class by efficient augmentation there has increase in accuracy of 1.24% on Hate Tweet Dataset by Random Forest Classifier while embedding with TF-IDF. On Spam Email Dataset there has been increase of 0.83% by Stochastic Gradient Descent Classifier while embedding with USE. On Yelp Reviews Dataset there has increase of 3.55% by Random Forest Classifier while embedding with TF-IDF.

Overall, the analysis is that only applying any random data augmentation on any certain dataset can tend to decrease in accuracy whereas encompassing ensembles of data augmentation leverages the benefit of selecting best efficient data augmentation technique for that dataset and works well in generalized way.

As we are also improvising the problem of class imbalance that's why we applied ensemble of text data augmentation on only on minority class. The future work of this study can include that when there is balanced small dataset then the full expansion of dataset using ensemble of text data augmentation can be done to improve the model performance. Apart from full expansion of dataset utilization of ensemble of sentence level augmentation techniques can also improvise the model performance by maintaining the semantics of data.

## VII. ACKNOWLEDMENTS

The author would like to acknowledge the institution, Symbiosis Institute of Technology, Pune for its relentless support and for providing an encouraging learning platform.

## REFERENCES

[1] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., &Kuksa, P. (2011). Natural language processing (almost) from scratch. Journal of machine learning research, 12(ARTICLE), 2493-2537.

[2] Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep learning--based text classification: a comprehensive review. ACM Computing Surveys (CSUR), 54(3), 1-40.

[3] Shorten, C., &Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. Journal of big data, 6(1), 1-48.

[4] Wei, J., & Zou, K. (2019). Eda: Easy data augmentation techniques for boosting performance on text classification tasks. arXiv preprint arXiv:1901.11196.

[5] Min, J., McCoy, R. T., Das, D., Pitler, E., &Linzen, T. (2020). Syntactic data augmentation increases robustness to inference heuristics. arXiv preprint arXiv:2004.11999.

[6] Wang, X., Pham, H., Dai, Z., &Neubig, G. (2018). SwitchOut: an efficient data augmentation algorithm for neural machine translation. arXiv preprint arXiv:1808.07512.

[7] Feng, S. Y., Gangal, V., Wei, J., Chandar, S., Vosoughi, S., Mitamura, T., &Hovy, E. (2021). A survey of data augmentation approaches for nlp. arXiv preprint arXiv:2105.03075.

[8] Liu, P., Wang, X., Xiang, C., & Meng, W. (2020, August). A survey of text data augmentation. In 2020 International Conference on Computer Communication and Network Security (CCNS) (pp. 191-195). IEEE.

[9] Bayer, M., Kaufhold, M. A., & Reuter, C. (2021). A survey on data augmentation for text classification. arXiv preprint arXiv:2107.03158.

[10] Belinkov, Y., & Bisk, Y. (2017). Synthetic and natural noise both break neural machine translation. arXiv preprint arXiv:1711.02173.

[11] Ningtyas, A. M., Hanbury, A., Piroi, F., & Andersson, L. (2021, December). Data Augmentation for Layperson's Medical Entity

Linking Task. In Forum for Information Retrieval Evaluation (pp. 99-106).

[12] Xie, Z., Wang, S. I., Li, J., Lévy, D., Nie, A., Jurafsky, D., & Ng, A. Y. (2017). Data noising as smoothing in neural network language models. arXiv preprint arXiv:1703.02573.

[13] Wei, J., & Zou, K. (2019). Eda: Easy data augmentation techniques for boosting performance on text classification tasks. arXiv preprint arXiv:1901.11196.

[14] Dai, X., & Adel, H. (2020). An analysis of simple data augmentation for named entity recognition. arXiv preprint arXiv:2010.11683.

[15] Peng, B., Zhu, C., Zeng, M., & Gao, J. (2021). Data augmentation for spoken language understanding via pretrained language models.

[16] Guo, H., Mao, Y., & Zhang, R. (2019). Augmenting data with mixup for sentence classification: An empirical study. arXiv preprint arXiv:1905.08941.

[17] Chen, J., Wang, Z., Tian, R., Yang, Z., & Yang, D. (2020). Local additivity based data augmentation for semi-supervised NER. arXiv preprint arXiv:2010.01677.

[18] Zhang, X., Zhao, J., &LeCun, Y. (2015). Character-level convolutional networks for text classification. Advances in neural information processing systems, 28.

[19] Wang, W. Y., & Yang, D. (2015, September). 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 (pp. 2557-2563).

[20] Liu, S., Lee, K., & Lee, I. (2020). Document-level multi-topic sentiment classification of email data with bilstm and data augmentation. Knowledge-Based Systems, 197, 105918.

[21] Marivate, V., &Sefara, T. (2020, August). Improving short text classification through global augmentation methods. In International Cross-Domain Conference for Machine Learning and Knowledge Extraction (pp. 385-399). Springer, Cham.

[22] Rizos, G., Hemker, K., & Schuller, B. (2019, November). Augment to prevent: short-text data augmentation in deep learning for hate-speech classification. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management (pp. 991-1000).

[23] Haralabopoulos, G., Torres, M. T., Anagnostopoulos, I., & McAuley, D. (2021). Text data augmentations: Permutation, antonyms and negation. Expert Systems with Applications, 177, 114769.

[24] Kashefi, O., & Hwa, R. (2020, November). Quantifying the evaluation of heuristic methods for textual data augmentation. In Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020).

[25] Madukwe, K. J., Gao, X., &Xue, B. (2022). Token replacement-based data augmentation methods for hate speech detection. World Wide Web, 1-22.

[26] Coulombe, C. (2018). Text data augmentation made simple by leveraging nlp cloud apis. arXiv preprint arXiv:1812.04718.

[27] Regina, M., Meyer, M., &Goutal, S. (2020). Text Data Augmentation: Towards better detection of spear-phishing emails. arXiv preprint arXiv:2007.02033.

[28] Louvan, S., &Magnini, B. (2020). Simple is better! lightweight data augmentation for low resource slot filling and intent classification. arXiv preprint arXiv:2009.03695.

[29] Palomino, D., & Luna, J. O. (2020). Palomino-Ochoa at TASS 2020: Transformer-based Data Augmentation for Overcoming Few-Shot Learning. In IberLEF@ SEPLN (pp. 171-178).

[30] Kobayashi, S. (2018). Contextual augmentation: Data augmentation by words with paradigmatic relations. arXiv preprint arXiv:1805.06201.

[31] Wu, X., Lv, S., Zang, L., Han, J., & Hu, S. (2019, June). Conditional bert contextual augmentation. In International Conference on Computational Science (pp. 84-95). Springer, Cham.

[32] Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., ... & Liu, Q. (2019). Tinybert: Distilling bert for natural language understanding. arXiv preprint arXiv:1909.10351.

[33] Hou, Y., Liu, Y., Che, W., & Liu, T. (2018). Sequence-to-sequence data augmentation for dialogue language understanding. arXiv preprint arXiv:1807.01554.

[34] Kang, D., Khot, T., Sabharwal, A., &Hovy, E. (2018). Adventure: Adversarial training for

textual entailment with knowledge-guided examples. arXiv preprint arXiv:1805.04680.

[35] Sennrich, R., Haddow, B., & Birch, A. (2015). Improving neural machine translation models with monolingual data. arXiv preprint arXiv:1511.06709.

[36] Yu, A. W., Dohan, D., Luong, M. T., Zhao, R., Chen, K., Norouzi, M., & Le, Q. V. (2018). Qanet: Combining local convolution with global self-attention for reading comprehension. arXiv preprint arXiv:1804.09541.

[37] Xie, Q., Dai, Z., Hovy, E., Luong, T., & Le, Q. (2020). Unsupervised data augmentation for consistency training. Advances in Neural Information Processing Systems, 33, 6256-6268.

[38] Luque, F. M. (2019). Atalaya at TASS 2019: Data augmentation and robust embeddings for sentiment analysis. arXiv preprint arXiv:1909.11241.

[39] Si, C., Zhang, Z., Qi, F., Liu, Z., Wang, Y., Liu, Q., & Sun, M. (2020). Better robustness by more coverage: Adversarial training with mixup augmentation for robust fine-tuning. arXiv preprint arXiv:2012.15699.

[40] Anaby-Tavor, A., Carmeli, B., Goldbraich, E., Kantor, A., Kour, G., Shlomov, S., ... &Zwerdling, N. (2020, April). Do not have enough data? Deep learning to the rescue!. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 34, No. 05, pp. 7383-7390).

[41] Fadaee, M., Bisazza, A., &Monz, C. (2017). Data augmentation for low-resource neural machine translation. arXiv preprint arXiv:1705.00440.

[42] Parida, S., &Motlicek, P. (2019, November). Abstract text summarization: A low resource challenge. 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) (pp. 5994-5998).

[43] Cai, H., Chen, H., Song, Y., Zhang, C., Zhao, X., & Yin, D. (2020). Data manipulation: Towards effective instance learning for neural dialogue generation via learning to augment and reweight. arXiv preprint arXiv:2004.02594.

[44] Lu, K., Mardziel, P., Wu, F., Amancharla, P., & Datta, A. (2020). Gender bias in neural natural language processing. In Logic, Language, and Security (pp. 189-202). Springer, Cham.

[45] Kafle, K., Yousefhussien, M. A., &Kanan, C. (2017, September). Data Augmentation for Visual Question Answering. In INLG (pp. 198-202).

[46] Huang, J., Li, Y., Tao, J., Lian, Z., Niu, M., & Yang, M. (2018, October). Multimodal continuous emotion recognition with data augmentation using recurrent neural networks. In Proceedings of the 2018 on Audio/Visual Emotion Challenge and Workshop (pp. 57-64).

[47] Kim, H. Y., Roh, Y. H., & Kim, Y. G. (2019, June). Data augmentation by data noising for open-vocabulary slots in spoken language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop (pp. 97-102).

[48] Ding, B., Liu, L., Bing, L., Kruengkrai, C., Nguyen, T. H., Joty, S., ... & Miao, C. (2020). DAGA: Data augmentation with a generation approach for low-resource tagging tasks. arXiv preprint arXiv:2011.01549.

[49] Wan, Z., Wan, X., & Wang, W. (2020, December). Improving grammatical error correction with data augmentation by editing latent representation. In Proceedings of the 28th International Conference on Computational Linguistics (pp. 2202-2212).

[50] Quteineh, H., Samothrakis, S., & Sutcliffe, R. (2020, January). Textual data augmentation for efficient active learning on tiny datasets. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 7400-7410). Association for Computational Linguistics.

[51] Smith, P. K., Mahdavi, J., Carvalho, M., Fisher, S., Russell, S., &Tippett, N. (2008). Cyberbullying: Its nature and impact in secondary school pupils. Journal of child psychology and psychiatry, 49(4), 376-385.

[52] Lee, P. (2016). Expanding the schoolhouse gate: Public schools (K-12) and the regulation of cyberbullying. Utah L. Rev., 831.

[53] Zhao, R., & Mao, K. (2017). Fuzzy bag-of-words model for document representation. IEEE transactions on fuzzy systems, 26(2), 794-804.

[54] Reimers, N., &Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamesebert-networks. arXiv preprint arXiv:1908.10084.

[55] Cer, D., Yang, Y., Kong, S. Y., Hua, N., Limtiaco, N., John, R. S., ... & Kurzweil, R. (2018). Universal sentence encoder. arXiv preprint arXiv:1803.11175.

[56] Khalid, M., Ashraf, I., Mehmood, A., Ullah, S., Ahmad, M., & Choi, G. S. (2020). GBSVM: sentiment classification from unstructured reviews using ensemble classifier. Applied Sciences, 10(8), 2788.