A Comparative study of Data-Augmentation Techniques for Imbalanced Hate speech data

Dhyankumar Shah
Lakehead University
Email: dshah26@lakeheadu.ca
Id : 1131701

Shruti Govani
Lakehead University
Email: sgovani@lakeheadu.ca
Id : 1143388

Abstract*— social media and microblogging apps allow people to share their information and express their personal view-points extensively and immediately. However, it also has some negative aspects such as hate speech. Recent advances in Natural Language Processing and Artificial Intelligence allow for more accurate detection of hate speech in textual streams. A significant challenge in this domain is that, while the presence of hate speech can be detrimental to the quality of service provided by social platforms, it still constitutes only a tiny fraction of the content available online, which can lead to performance deterioration due to majority class overfitting. To this end, we propose various data augmentation techniques with the goal of reducing class imbalance and maximizing the amount of information we can extract from our limited resources. After that, we apply them on a selection of top-performing deep architectures and hate speech datasets in order to classify them. The proposed approach outperforms all other considered algorithms. It achieves 0.69 F1-score for hate/non-hate classification.

Keywords: Hate Speech Detection, Data Augmentation, Binary Classification, Bert T5, Bert Contextual Embeddings.

I. Introduction

In recent era, social media is a golden tool for bringing people together and allowing them to network, share information, and express their personal viewpoints. Microblogging applications allow people all over the world to express and share their thoughts Immediately and extensively. However, every problem has multiple sides. It also has some negative aspects, such as hate speech, which is a crucial topic in the realm of social media. Hate speech is described as any form of communication that is abusive, insulting, intimidating, and/or incites violence or prejudice against an individual or a vulnerable group based on characteristics such as ethnicity, gender, sexual orientation, or religion [1]. These toxic comments and posts contain a variety of risks, including fake news, cyberbullies and online harassment which leads to various psychological issues for users, such as depression, frustration, and even suicidal thoughts. This thing occurs especially due to freedom and anonymity given to users and to the lake of effective regulations given by social media platforms. However, this toxic data can't manually censor, deleted, or controlled and thus we need a framework that can detect such data and might prevent publishing. Thus, several techniques and approaches have been introduced for the automatic detection of such contents using Artificial Intelligence and Natural Language Processing.

However, it requires a large amount of qualitative training data that are hard to be collected by human annotators. Text classification, on the other hand, has a lot of data imbalance problems. The problem of an unbalanced dataset occurs when data from one class is considerably greater or lower than data from other classes. Most machine learning and deep learning classification algorithms aren't designed to deal with unbalanced classes, and therefore tend to favor majority classes. To address these issues, many data augmentation approaches are offered in this paper. Data augmentation is a technique that can help generate a large amount of data from a small amount. The most commonly used data augmentation technique in NLP is very simple, relying on simply replacing words with synonyms or deleting words here and there for each text [2]. However, this approach is typically effective only when the original dataset is small, implying that it has no effect when the original dataset is already large. Furthermore, the structure of the text generated by this method is nearly unchanged. This has two unfavorable consequences. First and foremost, this usually has a minor impact, if any, on classification accuracy. Furthermore, the classifier trained on these generated data is prone to overfitting. In this paper we performed comparative analysis of various data augmentation techniques we have used so far.

The remainder of this paper goes as follows: In Section II, we introduce some related work. In Section III, we presented data pipeline of our work, then in section IV,we describe the dataset and proposed framework Also, we discussed the experiments which we conducted. In Section V, we represent and discuss the results of the classification accuracy. Finally, in Section VI, we conclude the page.

II. Literature Review

Automatic detection of hate speech has received considerable attention from natural language processing (NLP) research communities. Interest in this field has increased with the proliferation of social media and social platforms. [3]

There have been so many traditional models [4] [5] [6] and Deep learning models [7][8] have been developed to detect hate speech in social media. However, the accuracy of sentiment analysis and hate speech is likely to be reduced due to insufficient and imbalanced training data.

Furthermore, an increase in training data does not always result in a solution to the learning problem. Nonetheless, the quality of a supervised classifier is still determined by the data. Many different methods for artificially creating such data, known as data augmentation, have emerged from the field of computer vision. [9]

A commonly used data augmentation technique in NLP is randomly replacing or deleting words. In this paper, [1] authors used a synonym dictionary to augment data for text classification. The amount of data is increased by replacing, inserting, swapping, and deleting words in each text at random. However, it demonstrates that when the number of original data points exceeds 2,000, the effect becomes worse. Thus, this approach is typically effective only when the original dataset is small, implying that it has no effect when the original dataset is already large. The main reason is that these operations performed on words did not consider the sentiment information.

Rizos et al. [10] presented three text data augmentation techniques and tested their effectiveness on three databases using four top performing deep models for hate speech detection. The first method is based on synonym substitution and word embedding vector closeness. The second one uses word token warping along the padded sequence, and the last one deals with class-conditional, recurrent neural language generation.

In this paper, [11] the author examines the performance of various machine learning models in the classification of toxic comments and proposes an ensemble approach called RVVC (Regression vector voting classifier). RVCC is an ensemble classifier that combines the logistic regression and support vector classifier through soft voting criteria. The results show that models perform poorly on the imbalanced dataset, whereas the balanced dataset improves classification accuracy. For that, the authors have used SMOTE (Synthetic minority oversampling technique) and random under-sampling technique. However, these methods don't add any new information to the model instead new examples get synthesized from the existing samples.

Generative based methods employ deep generative models such as GANs and VAEs, generate sentences from continuous from a continuous space with desired attributes of sentiments and tense. [12] However, it is quite challenging to guarantee the quality of sentences generated using these algorithms in terms of label compliance and sentence readability.

Back translation and Paraphrasing are another technique used for data augmentation in order to ensure class balances and embedded based deep networks to yield a binary classification [13]. Particularly, Back translation, is based on a transformer architecture that includes an encoder decoder model. The paraphrasing method employs machine translation models in which text is translated to an intermediate language and then back to the original.

In this paper [14] author used deep bidirectional transformer architecture to produce pre-trained context-dependent embeddings.And, it has proven to be powerful in solving many NLP tasks and, it also appears to handle imbalanced classification well.

we proposed Bert Contextual Embeddings to handle data imbalance for Hate Speech dataset. The new sentences are generated by stochastically selecting words and replacing them with other words without changing the context using powerful transformer model (BERT). The generated hate speech is then used to enhance data to improve hate speech recognition automatically.

III. Data Pipeline

Here, In this paper we proposed a Data Augmentation framework to improve the accuracy of sentiment analysis.In the proposed framework, a given training dataset is relatively small in size, thus we used data augmentation methods to generate artificial data. The original training set is then combined with the augmented set to train the classifier,To assess the contribution of newly generated data to classification accuracy, we train another classifier solely on the original data and compare the two classifiers on the validation/test set.

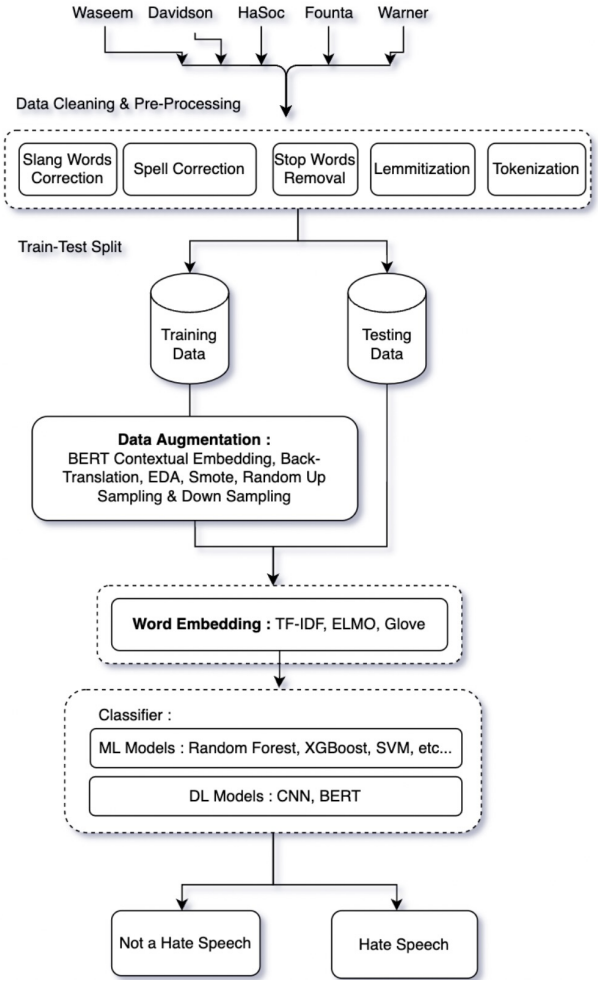

Fig. 1. Data Pipeline for Hate Speech Detection

IV. Methods and Technologies

A. Data-Set

Five separate datasets from the local repositories were merged together for this analysis. Although all of the datasets represent hate and non-hate categorization data, each dataset's general category varies. Davidson Dataset, for instance, is based on Offensive and Non Offensive. Both Warner and Waseem are based on racism. The table below displays the distribution of hate and non-hate in these data sets:

Name of the Dataset	Total	Hate	Non-Hate
Warner (14.65% - 85.34%)	1269	186	1085
Waseem (3.49% - 96.5%)	14142	494	13648
Founta (14.21% - 85.7%)	59319	8434	50885
Hasoc (50% - 50%)	3708	1856	1852
Davidson (5.77% - 94.22%)	24783	1430	23353
Total (12.01% - 87.9%)	103221	12400	90823

B. Data Cleaning and Pre-processing Techniques

1) Preprocessing using Regular Expression: Using Regular Expression by pattern matching we removed hashtags, video and audio tags, mentions, unnecessary spaces, digits from the tweets.

2) Stop words removal: Stop words are the words that are often excluded before processing natural language. These are the most widespread words in any language (such as articles, prepositions, pronouns, conjunctions, etc.), yet they do not significantly add to the text's content. The words "the," "a," "an," "so," and "what" are a few examples of stop words in English.**[https://towardsdatascience.com/text-pre-processing-stop-words-removal-using-different-libraries-f20bac19929a]**

3) Spell Correction: In this step, we detected and corrected spelling errors from the tweets using Autocorrect library.

4) Contraction Map (Expansion of Short Forms): In English contractions, we often drop vowels from a word to form the contractions. For example, I'll be there in 2 mins. Are u not gng there? I'd like to see u near d park. However, Removing contractions contributes to text standardization and is useful when working on twitter data , or on reviews as the words play an important role. Here, we created our own contraction map dictionary for this purpose.

5) Tokenization: Tokenization is the process of breaking down a large chunk of text into smaller tokens. Tokens can be words, characters, or subwords in this case. Tokenization may therefore be divided into three categories: word, character, and subword tokenization.
For eg : "Natural Language Processing."
The most popular method for creating tokens is to use space. Using space as a delimiter, the tokenization of the text produces three tokens: Natural-Language-Processing.[15]

6) Lemmitization: It is the act of putting together the inflected elements of a word into a single unit known as the word's lemma or vocabulary form. This is similar to stemming, except that it gives meaning to certain words. It unites content that has similar meanings to a single word, in simple terms.
It is defined as an algorithm technique for locating a root word's lemma rather than a root stem's lemma. It's determined by the word's intended meaning.
Example:
1. rocks: rock
2. Corpora: corpus
3. Better: good [16]

C. Feature Extraction and Development

1) TF-IDF Vectorizer: The phrase TF-IDF stands for term frequency-inverse document frequency. The TF-IDF weight is a statistical metric for determining how essential a word is to a text in a corpus or collection. The importance of a word rises in direct proportion to the number of times it appears in the text, but is counterbalanced by its frequency in the corpus.

- Term Frequency (TF) : It's a count of how often a term appears in the current document. Because every document is varied in length, a phrase may appear significantly more frequently in large papers than in shorter ones. To normalise, the word frequency is frequently divided by the document length.

$$TF(t) = \frac{Number\ of\ times\ term\ t\ appears\ in\ a\ document}{Total\ number\ of\ terms\ in\ the\ document}$$

Fig. 2. Term Frequency

- Inverse Document Frequency (IDF) : It's a grading system that determines how uncommon a word is throughout papers. The IDF is a metric for determining how uncommon a phrase is. The higher the IDF score, the rarer the phrase.

$$IDF(t) = log_e(\frac{Total\ number\ of\ documents}{Number\ of\ documents\ with\ term\ t\ in\ it})$$

Fig. 3. Inverse Document Frequency

Thus,

$$TF - IDF score = TF * IDF$$

Fig. 4. TF-IDF Equation

2) GLOVE Embedding:

- Word embedding is a way where in each word in the text is represented by a vector with real values in a high dimensional space. The vectors are learned such that similar representation will be present for words that have similar meanings in the vector space.
- Glove is a semantic vector models of language represent each word with a real valued vector.
- A vocabulary of 400 thousand words and a data set of one billion words were used to train the GloVe embedding algorithm. The sizes of the embedding vectors range from 50 to 200 to 300 dimensions.
- Glove model uses word frequency and global co-occurrence count matrix.
- By effectively doing dimensionality reduction on the co-occurrence counts matrix, count-based models learn their vectors.
- These vectors can be used as features in a variety of applications such as information retrieval, document classification, question answering , NER (Named entity reorganization and parsing).[26]

3) ELMO Embedding:

- ELMo stands for Embeddings from Language Models, and hence it also has the ability to predict the next word in a sentence.
- It is a method of presenting a deeply contextualised word that replicates both the intricate details of word usage and the ways in which these uses change depending on the linguistic setting.

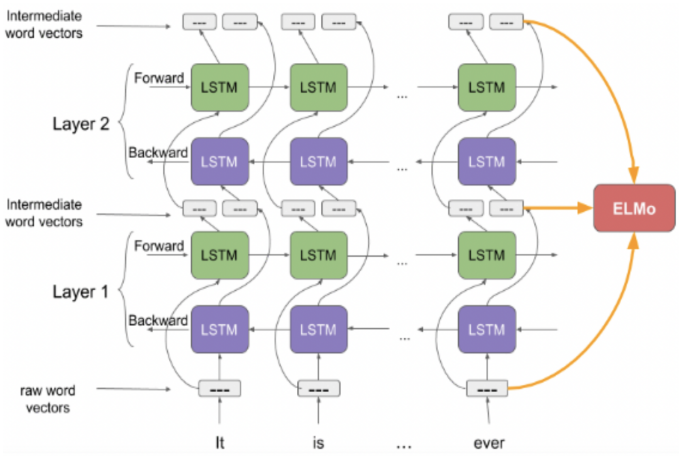

Fig. 5. ELMO Embedding

- So, Instead of using a fixed embedding for each word, like models like GloVe do , ELM) looks at the entire sentence before assigning each word in it its embedding.
- Elmo employs bi-directional LSTM in its training process to ensure that its language model comprehends both the word that comes after it in the phrase as well. A two-layer, bidirectional LSTM backbone is present. The second layer is inserted between the first layer and the residual connection..[25]

D. Data Augmentation Techniques

Distinct data augmentation approaches for textual data are divided into different groups in the figure as follows.

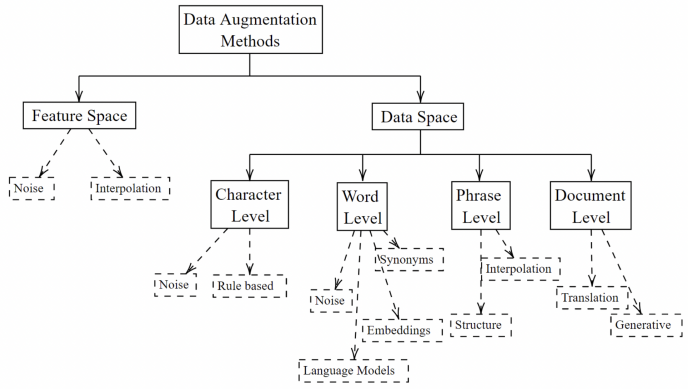

Fig. 6. Division of Data Augmentation Techniques[17]

1) EDA - Easy Data Augmentation Techniques:

- Synonym Replacement (SR): Pick n words from the phrase that aren't stop words at random. Each of these words should be replaced with a synonym picked at random.
- Random Insertion (RI): Identifying and extracting synonyms for certain randomly picked words in the phrase that are not StopWords. Using this determined synonym in a phrase at a random location.

Operation	Sentence
None	A sad, superior human comedy played out on the back roads of life.
SR	A ***lamentable***, superior human comedy played out on the ***backward*** road of life.
RI	A sad, superior human comedy played out on ***funniness*** the back roads of life.
RS	A sad, superior human comedy played out on ***roads*** back ***the*** of life.
RD	A sad, superior human out on the roads of life.

Fig. 7. Easy Data Augmentation Techniques

- Random Swap (RS): Randomly choosing two words in the sentence and swap their positions.
- Random Deletion (RD): Randomly removing word in the sentence.[18]

2) Back Translation:

- Temporary translation: Every single training set's original labelled data should be translated into a new language. It will be from English to French in our situation.
- Back translation: Translate each of the translated data sets back into their original languages, such as a French to English translation.
- Duplicate removal: At the conclusion of the procedure, we will obtain the matching back-translation for each original text data. The idea is to just maintain one instance of

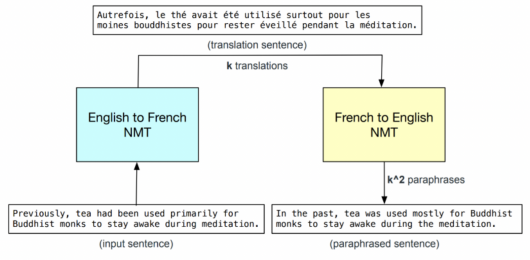

Fig. 8. Translation

each duplicate. Because of its intrinsic label preservation and very important paraphrase capabilities, the technique seems promising.

3) Smote-ENN Edited Nearest Neighbour: The capacity of ENN to remove certain observations from both classes that are determined to have a different class from the observation's class and its K-nearest neighbour majority class is combined with the ability of SMOTE to provide synthetic instances for the minority class in this technique.

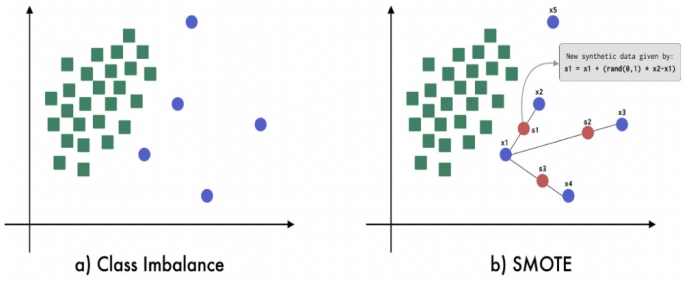

Fig. 9. Smote-ENN

- Select arbitrary information from the minority class.
- Identify the k closest neighbours of the random data and determine their distance from them.
- As a synthetic sample, multiply the difference by a value at random between 0 and 1, then add the result to the minority class.
- K is the number of neighbours that are closest to you. Unless otherwise specified, K=3.
- Among the other observations in the dataset, locate the observation's K-nearest neighbour. Then, retrieve the observation's majority class from the K-nearest neighbour
- The observation and its K-nearest neighbour are removed from the dataset if the class of the observation and the majority class from the observation's K-nearest neighbour are different.
- Repeat steps 2 and 3 as necessary to get the required percentage of each class.[27]

4) BERT Contextual Embedding:

- We assume an invariance that sentences are natural even if the words in the sentences are replaced with other words with paradigmatic relations.

Original	The quick brown fox jumps over the lazy dog
Synonym (PPDB)	The quick brown fox climbs over the lazy dog
Word Embeddings (word2vec)	The easy brown fox jumps over the lazy dog
Contextual Word Embeddings (BERT)	Little quick brown fox jumps over the lazy dog

Fig. 10. BERT Contextual Embedding

- At the word places, we stochastically swap out words with others that a bidirectional language model predicts. There are many context-sensitive terms, but they are all acceptable for enhancing the original language.[24]

E. Model Used

1) Machine Learning Models:

- Random Forest :
  hyperparameters used n_estimators = 100, criterion = 'gini
- XGBoost:
  Hyperparameters used n_estimators = 1500, eta = 0.05
- SVM:
  Hyperparameters used C = 0.028, multi_class = "crammer_singer", max_iter = 1000

TABLE I
RESULTS USING HYPER-PARAMETER TUNING

Model	Accuracy	F1	Recall	Precision
Random Forest	0.915	0.691	0.698	0.685
XGBoost	0.903	0.671	0.772	0.626
SVM	0.911	0.680	0.69	0.668

2) BERT: Language models often interpret the input sequence in one of two directions: left to right or right to left. When the goal is to predict/generate the next word, this type of one-directional training works effectively.
BERT, on the other hand, employs bidirectional training to get a better understanding of linguistic context. It's also known as "non-directional" at times. As a result, it considers both the preceding and subsequent tokens at the same time. BERT learns text representations by using Transformer's bidirectional training to language modelling.

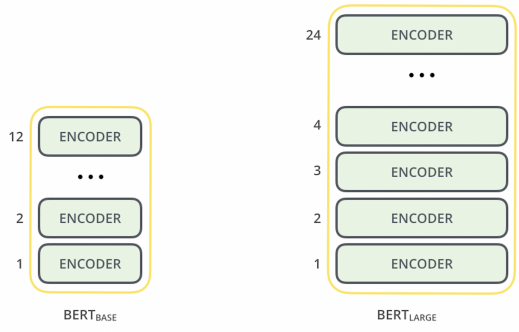

Fig. 11. Bert base and Bert large Model

There are two general BERT versions that have been pre-trained: The large model is a 24-layer, 1024-hidden, 16-heads, 340M parameter neural network architecture,

whereas the base model is a 12-layer, 768-hidden, 12-heads, 110M parameter neural network architecture.[20]

3) LSTM: Long-Short Term Memory (LSTM) is an acronym for Long-Short Term Memory. In terms of memory, LSTM is a sort of recurrent neural network that outperforms standard recurrent neural networks. LSTMs perform significantly better when it comes to memorising specific patterns. LSTM, like any other NN, can have several hidden layers, and as it goes through each layer, the relevant information is retained while the irrelevant information is eliminated in each cell [20].
We may use a variety of LSTMs for text categorization such as:

- BiLSTM
- Stacked LSTM
- GRU (Gated Recurrent Unit)
- BGRU (Bidirectional Gated Recurrent Unit)

V. Results

The classification report is represented here in the table II and table III. The results shown in table II are without data augmentation. And, the table III represents the results with data augmentation. We tried 4 data augmentation methods with various classifiers and word embeddings. And according to the table III, the combination of Random forest with BERT Contextual Embeddings gives highest F1-Score for class 1.

TABLE II
Results without Data Augmentation

Without Text Augmentation	F1	Recall	Precision
TF-IDF + RF	0.69	0.61	0.81
TF-IDF + XGBoost	0.68	0.59	0.79
TF-IDF + SVM	0.67	0.58	0.77
BERT Pooled + BERT T5	0.2	0.14	0.37
GLOVE + CNN	0.52	0.45	0.61
ELMO + CNN	0.56	0.46	0.66

TABLE III
Results with Data Augmentation

With Text Augmentation	F1	Recall	Precision
TF-IDF + BERT + RF	**0.69**	**0.71**	**0.68**
TF-IDF + BERT + XGBOOST	0.69	0.7	0.67
TF-IDF + BERT + SVM	0.63	0.66	0.6
ELMO + BERT + CNN	0.58	0.49	0.67
GLOVE + BERT + CNN	0.55	0.51	0.6
TF-IDF + EDA + RF	0.43	0.3	0.73
TF-IDF + EDA + SVM	0.46	0.37	0.6
TF-IDF + BT + RF	0.67	0.59	0.85
TF-IDF + BT + XGBoost	0.66	0.59	0.8
TF-IDF + BT + SVM	0.62	0.55	0.71
TF-IDF + SMOTE + RF	0.24	0.97	0.3
TF-IDF + SMOTE + SVM	0.24	0.96	0.14

The original dataset has only 12.01% of hate speech data. However, We increased the minority class until the threshold point (50%), using Bert contextual embeddings. The Fig.12 represents the graphical representation of the classification report.

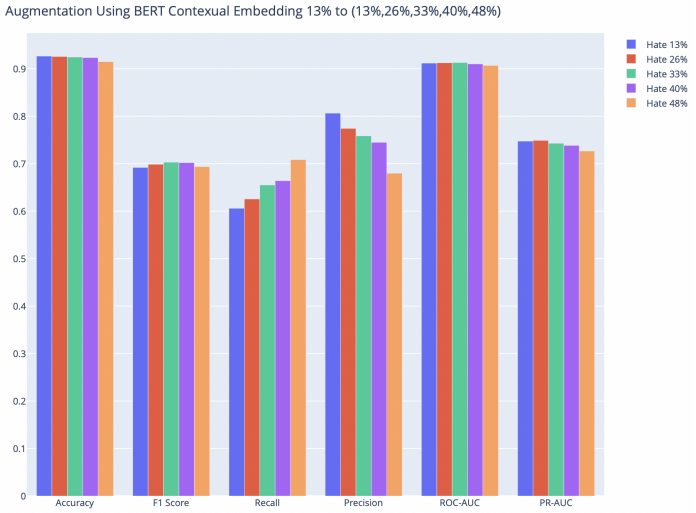

Fig. 12. Augmentation using BERT Contextual Embedding

VI. Conclusion

According to our research comparing various model types, word embedding techniques, and augmentation approaches, we have come to the conclusion that augmentation is a critical element of an unbalanced dataset to get a better F1 score. BERT contextual embedding is highly helpful for text augmentation in our scenario since it doesn't alter the sentence's main idea; rather, it only adds supporting words. Our tests reveal that the combination of TD-IDF + BERT Contextual Embedding + Random Forest provides the greatest F1-Score, Recall, and accuracy.

VII. References

[1] Chiril, Patricia, et al. "Emotionally Informed Hate Speech Detection: A Multi-Target Perspective - Cognitive Computation." SpringerLink, Springer US, 28 June 2021, https://link.springer.com/article/10.1007/s12559-021-09862-5Tab

[2] J. Wei and K. Zou, "EDA: Easy data augmentation techniques for boosting performance on text classification tasks", Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP), pp. 6382-6388, 2019.

[3] Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In Proceedings of the Fifth International Workshop on Natural Language Processing for social media, pages 1–10.

[4] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In Eleventh international aaai conference on web and social media.

[5] Ying Chen, Yilu Zhou, Sencun Zhu, and Heng Xu.

2012. Detecting offensive language in social media to protect adolescent online safety. In 2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Conference on Social Computing, pages 71–80. IEEE.

[6] Zeerak Waseem. 2016. Are you a racist or am I seeing things? annotator influence on hate speech detection on twitter. In Proceedings of the first workshop on NLP and computational social science, pages 138–142.

[7] Antigoni Maria Founta, Despoina Chatzakou, Nicolas Kourtellis, Jeremy Blackburn, Athena Vakali, and Ilias Leontiadis. 2019. A unified deep learning architecture for abuse detection. In Proceedings of the 10th ACM Conference on Web Science, pages 105–114.

[8] Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. 2018. Long text generation via adversarial training with leaked information. In Thirty-Second AAAI Conference on Artificial Intelligence.

[9] Bayer, Markus, et al. A Survey on Data Augmentation for Text Classification. https://arxiv.org/ftp/arxiv/papers/2107/2107.03158.pdf.

[10] G. Rizos, K. Hemker, B. Schuller, 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, 2019, pp. 991–1000.

[11] Rupapara et al. "Impact of Smote on Imbalanced Text Features for Toxic Comments Classification Using RVVC Model." IEEE Xplore, ieeexplore.ieee.org/document/9440474.

[12] Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, pages 10–21.

[13] Edunov S., Ott M., Auli M., Grangier D. Understanding back-translation at scale Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium (2018), pp. 489-500, 10.18653/v1/D18-1045

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[15] "What is tokenization: Tokenization in NLP," Analytics Vidhya, 23-Jul-2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/05/what-is-tokenization-nlp/. [Accessed: 04-Mar-2022].

[17] "An interpretation of lemmatization and ... - researchgate.net." [Online]. Available: https://www.researchgate.net/profile/Siddhartha-B-S/publication/348306833 An Interpretation of Lemmatization and Stemming in Natural Language Processing/links/6048467f299bf1e078696a3a/An-Interpretation-of-Lemmatization-and-Stemming-in-Natural-Language-Processing.pdf. [Accessed: 04-Mar-2022].

[18] D. M. Blei, "Probabilistic topic models," Communications of the ACM, vol. 55, no. 4, pp. 77–84, 2012.

[19] "Basic introduction to the bag of words model in 6 points," Jigsaw Academy, 08-Mar-2021. [Online]. Available:https://www.jigsawacademy.com/blogs/ai-ml/bag-of-words. [Accessed: 04-Mar-2022].

[20] J. D'Souza, "An introduction to bag-of-words in NLP," Medium, 04-Apr-2018. [Online]. Available: https://medium.com/greyatom/an-introduction-to-bag-of-words-in-nlp-ac967d43b428. [Accessed: 04-Mar-2022].

[21] Z. Ali, "A simple word2vec tutorial," Medium, 07-Jan-2019. [Online]. Available: https://medium.com/@zafaralibagh6/a- simple-word2vec-tutorial-61e64e38a6a1. [Accessed: 04-Mar-2022].

[22] "Bert: Bert Transformer: Text classification using bert," Analytics Vidhya, 29-Jun-2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/06/why-and-how-to-use-bert-for-nlp-text-classification/. [Accessed: 04-Mar-2022].

[23] "LSTM for text classification: Beginners Guide to Text Classification," Analytics Vidhya, 30-Jun-2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/06/lstm-for-text-classification/. [Accessed: 04-Mar-2022]

[24] Kobayashi, S., 2022. Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations. [online] Available at: ¡https://arxiv.org/pdf/1805.06201v1.pdf¿ [Accessed 29 July 2022].

[25] Medium. 2022. Learn how to build powerful contextual word embeddings with ELMo. [online] Available at: ¡https://medium.com/saarthi-ai/elmo-for-contextual-word-embedding-for-text-classification-24c9693b0045: :text=ELMo

[26] Medium. 2022. Light on Math ML: Intuitive Guide to Understanding GloVe Embeddings. [online] Available at: ¡https://towardsdatascience.com/light-on-math-ml-intuitive-guide-to-understanding-glove-embeddings-b13b4f19c010¿ [Accessed 29 July 2022].

[27] Medium. 2022. Imbalanced Classification in

Python: SMOTE-ENN Method. [online] Available at: ¡https://towardsdatascience.com/imbalanced-classification-in-python-smote-enn-method-db5db06b8d50¿ [Accessed 29 July 2022].