HATE SPEECH DETECTION

1 Introduction

In the more recent years, as social media usage has been increasing among people, hate speech has furthermore been drastically increasing across all social media platforms, especially on Twitter [1]. There is no direct definition for hate speech, however, it is agreed upon that it is the use of derogatory terms and discriminatory language against certain groups of people based on their characteristics or where someone is from [2] e.g., religion, ethnicity, nationality, race, colour, sexuality, and gender. This eventually leads to intolerance and unfair bias in society which similarly leads hate crime and physical violence against the targeted groups. Hate speech online is an enormous problem in society, recent researchers have been trying to find ways to be able to detect hate speech [3].

However, research has come to convey that it is a difficult task detecting hate speech online, last year alone 6000 tweets were posted on twitter every second [4]. The enormous amount of data cannot possibly be moderated by any reasonable number of people. This is mainly because such efforts based on manual labour to comb through and identify offensive materials is rigorous, waste of time, and not viable or accessible. The task for using natural language processing and machine learning techniques to moderate tweets for us has been developing for some time.

The task aims to identify hate on Twitter using the dataset provided by SemEval-2019 Task. As the given training dataset is small, transfer learning and data augmentation are explored in our work. To fine-tune the model, hyperparameters are optimized using Keras Tuner. The final model was able to perform better than the baseline models and the performance metrics were as good as the second-best team that participated in the SemEval 2019 Task 5 named *HatEval*. The details of this competition are available at *www.competitions.codalab.org* [5].

This report will take the following structure. Section 1 will present and overview a performance summary of similar work. Section 3 contains a description of the dataset. Section 4 describes the implemented methodology. Section 5 discusses experimental settings and the reason for their used. Section 6 then presents the results of the trained model on the test data. Section 7 analyses the performance of our model. Finally, section 8 summaries the projects deliverables and discusses potential improvements to the techniques used here.

2 Literature Review/ Related Work

Many studies have used different models all with a variety of performance. The use of various neural networks architectures such as Convolutional Neural Networks (CNNs) for sentence classification [6] and Recurrent Neural Networks (RNNs) [7] has become very popular in Natural Language Processing. A recent and successful paper used an ensemble of recurrent neural networks to distinguish between racism and sexism with a corpus of 16k tweets [8]. The paper acquired a F-score of 0.932 for their ensemble of classifiers. The best performance of the papers discussed in this paper. However, this method is very computationally expensive as it uses multiple neural network classifiers to create the ensemble. Given the scope of this project it is unrealistic to expect similar performance. More traditional machine learning algorithms have also seen use within hate speech detection. Badjatiya P et al. (2017) [9] experimented with multiple classifiers Logistic Regression, Random Forest, SVMs, Gradient Boosted Decision Trees (GBDTs) and Deep Neural Networks (DNNs) and found that Deep Neural Networks outperformed other models in hate speech detection task. Indurthi et al. (2019) [10] designed a model using sentence embeddings for transforming the input and SVM (with RBF kernel) for hate speech detection.

Twitter contains many features other than the just the tweet that can be used to inform the machine of possible hate speech. Unsvag and Gamback (2018) [11] investigated the effect that features available on twitter effects efforts to develop a hate speech detection model. These features included twitter profiles gender, the profiles follow/follower network and characteristics on user profiles (these include presence of pictures, personal information on display and location). The finding of this work was that taking these features into account only slightly lead to better classification performance if any.

Other social media data has been used to detect hate speech, Pratiwi et al. (2018) [12] used some traditional machine learning techniques on Instagram comments in

Indonesian. The highest performance acquired was F1= 65.7% when using FastText. This work shows how models preform differently in different languages, and a model that is best for one language is not necessarily the best for another. Given a well optimized method a neural network will put preform traditional techniques as evident form the papers discussed here.

The model presented here is very much related to the work done by Montejo-Raéz et al. (2019) [13]. They proposed data augmentation with paraphrasing tools and transfer learning for identifying hate speech in tweets. The transfer learning was done by allowing the pre-trained weights to be retrained by the model during training process. But in our model, we created a word2vec model to build vocabulary specific to the given dataset and then added the GloVe vocabulary and initial weights to retrain it. We did not allow the neural network model to change these weights. With this method, the model was able to perform comparatively better.

3 Description of the Dataset

The provided datasets consist of 9000 twitter messages for training, 1000 messages for validation and 2971 messages for testing. All the datasets were almost balanced with 42% of the messages labelled as hate speech. Some patterns could be seen in the most frequent words used in the hate speech texts. The hashtags and user-mentions like 'buildthewall', '@realdonaldtrump' are very frequent in the hate speech texts as shown on figure 1 below.

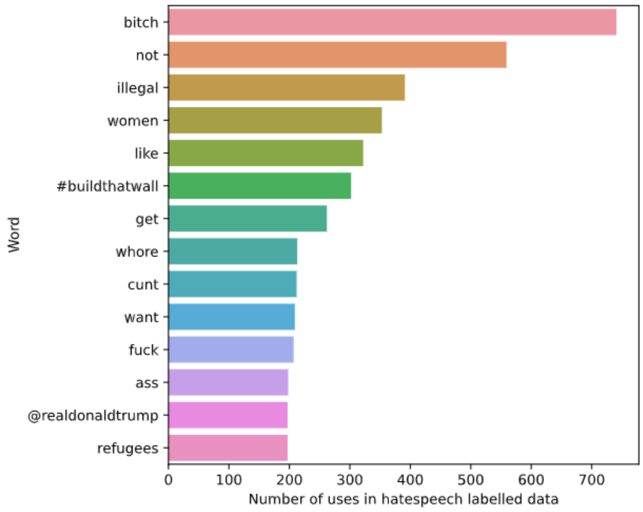


Figure 1: Histogram of the most used words within the hatespeech labelled training data. The stop words have been removed from the data.

T-test was conducted to investigate if the average number of characters in texts labelled as 'Hate Speech' is significantly different from the average number of characters in 'Not Hate Speech' texts. The obtained p-value was 0.003. This very small value means that there is a statistically significant difference in the two distributions. We can reject the null hypothesis that the two distributions are the same, knowing that there is only a 0.3% probability that this difference is due to random gaussian sampling. The distribution of data is different in training and test data set which made it difficult to get good model performance on test data set. This can be seen below in figure 2.

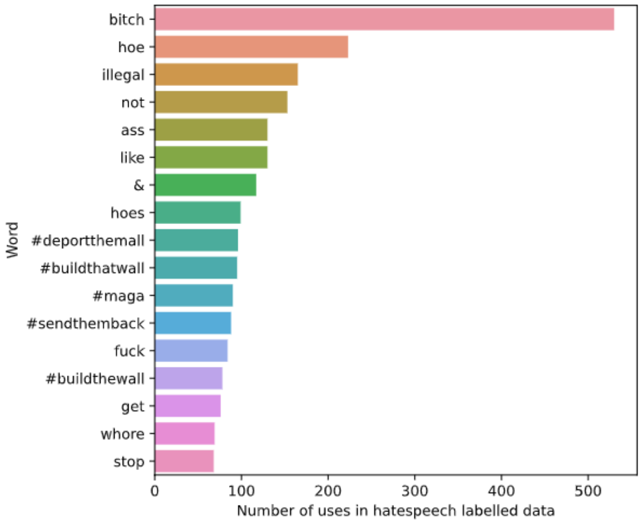


Figure 2: Histogram of the most used words in the test data set. The stop words have been removed from the data.

The most noticeable difference is the prevalence of the word "hoe" in the test data set. This word is the second most frequent in this data set but is absent from the top words the training data. The other words are also mostly ranked differently, and their share of the distribution is lower than for the training set. So, most words will occur less frequently in the test data set. This will lead to reduced performance for the model, but this problem could be overcome with a larger and more representative training data set.

As the models using neural networks can perform well with minimum data pre-processing, we experimented with and without stop words and punctuation removal. But for this dataset, removing them gave better results.

The texts have been pre-processed as follows:

1. Texts are converted to lower case.
2. URLs, HTML references, user-mentions and numbers are removed.
3. NLTK TweetTokenizer is used to tokenize texts so that the emojis and hashtags are kept together.
4. Punctuations and stop words are removed after tokenization.
5. Texts are updated with lemma form of words.

4 Methodology

One of the challenges we faced was the small amount of training data, which did not reflect the true population rate of hateful tweets compared to non-hateful, thus the deep neural network trained on this training dataset was overfitting and was not able to perform well on test data. We found that embedding layer created based on the training data vocabulary always resulted in overfitting even after including dropouts with high probability.

Hence to tackle these problems, we experimented with transfer learning using *GloVe embeddings* and *data augmentation* technique proposed by Wei and Zou (2019) [14].

The neural architecture is implemented using Keras library on TensorFlow. Initially a word2vec embedding model was created which gave 200 - dimension vector for each word in the texts. This model was retrained using the pre-trained weights from the GloVe twitter model provided by the Stanford NLP group [15] which is built over 2 billion tweets. This transfer learning approach outperformed other experimental settings. These included creating embeddings from the training data and embedding only with GloVe pre-trained weights. As expected, performance of word2vec and FastText word embedding models trained using small dataset was not good enough.

The sequence of layers used in the model is as follows:

1. Embedding layer with the weights from the model described above. This converts each word in texts into 200-dimension vector. Parameter – Trainable is set to False as we do not want the network to change the vector representation during the training.
2. A bi-directional LSTM recurrent network with 288 activations.
3. Dropout with the probability of 0.2.
4. A dense network with 224 activations and the ReLU function as activation function.
5. Classification layer with 1 activation using Sigmoid function.

Since bidirectional LSTM contains two LSTM layers running in the opposite directions and can understand the context better, it could perform better than the unidirectional LSTM. Hyperparameters were fine-tuned using Keras Tuner library.

Another technique we explored was increasing the number of training data samples as per the data augmentation technique suggested by Wei and Zou (2019) [14]. Where we apply 4 different operations on the data, these being synonym replacement, random insertion, random swapping, and random deletion. This technique was proposed mainly for boosting performance of text classification tasks. The instructions to increase samples are publicly available at Wei (2021) [16]. The sample size in training data was increased to 90,000. The neural network architecture used for training this new training data is same as the one used with transfer learning approach. However, this model could not perform well on test data, mainly because of the large number of unrealistic tweets created by this technique.

5 Experimental Setting

Various window sizes of 5, 10, 15 and 20 were tried for word2vec embedding. Levy and Goldberg (2014) [17] suggested that the larger windows tend to capture more topic/domain specific information. For our dataset, window of 20 gave better model performance than other settings. To understand if transfer learning helped to improve the performance, we compared the results with the model built using fixed weights from GloVe embeddings (not trainable). As shown in Table 1, transfer learning model performed better.

EMBEDDING	p	r	$F1$	$Accuracy$
TRANSFER LEARNING – GloVe	0.62	0.63	0.62	0.63
Weights from glove embeddings only	0.60	0.56	0.49	0.52

Table 1: Performance evaluation with and without transfer learning.

The hyperparameters such as number of activations in bi-directional LSTM recurrent network, dropout probability, activation function and number of activations in the dense network are fine-tuned using a Bayesian optimization method provided by Keras Tuner. The Bayesian optimizer chooses the first few hyperparameters randomly and then based on its performance it chooses the next best possible hyperparameter. The Loss function used for training the model was binary cross-entropy with Adaptive Moment (Adam) Optimization algorithm. The best learning rate for Adam was found using Keras Tuner Bayesian optimization method, as before. The number of epochs is controlled by the early stopping mechanism where parameter patience is set to 5 so that it waits for 5 more epochs before stopping if there is no progress on validation data set. As shown in figure 3, loss reduced gradually for validation and training datasets after a few epochs.

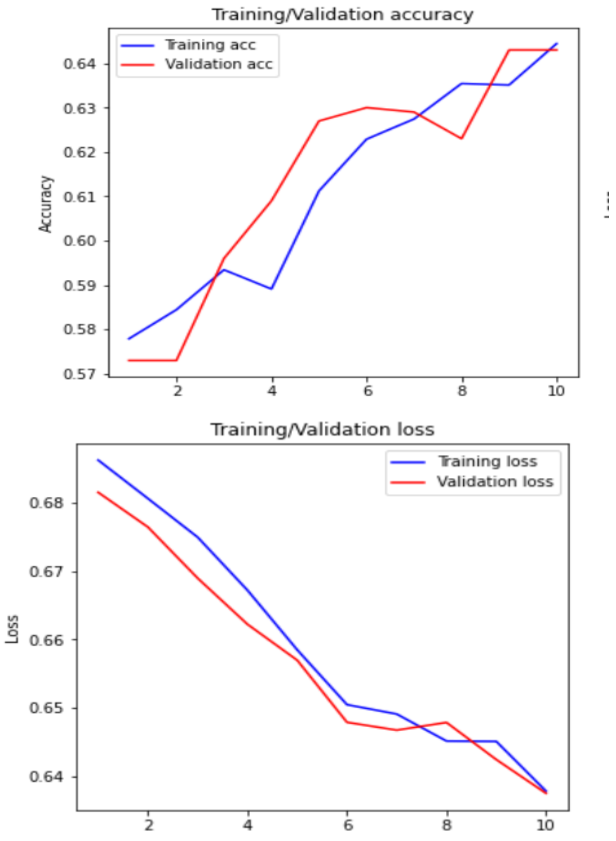


Figure 3: Plots showing accuracy (first plot) and loss (second plot) for each epoch for training and validation datasets.

6 Results

SVC baseline and MFC baseline models were created as given in the SemEval-2019 Task 5 [18]. The MFC baseline is a trivial model that assigns the most frequent label, estimated on the training set, to all the instances in the test set. The SVC baseline is a linear Support Vector Machine (SVM) based on a TF-IDF representation, where the hyper-parameters are the default values set by the scikit-learn Python library. As shown in Table 2, our model performed better than the baseline models with respect to all the metrics on the test dataset.

MODEL	***p***	***r***	***F1***	***Accuracy***
MFC BASELINE	0.37	0.50	0.37	0.58
SVC BASELINE	0.45	0.55	0.44	0.49
BIDIRECTIONAL LSTM MODEL	0.62	0.63	0.62	0.63

Table 2: Performance evaluation of the model and the baselines. Here p, r and F1 represent the macro average values of precision, recall and F1 respectfully.

7 Error Analysis

The confusion matrix obtained for the model is presented in the figure 5. The misclassification rate for Not hate speech is 35% and for hate speech it is 44%. This indicates that the model performed slightly better in categorizing Not hate speech than hate speech texts. The overall accuracy is 61%.

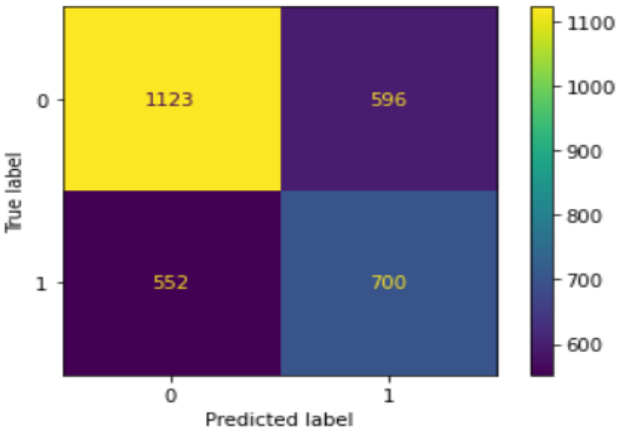


Figure 4: Confusion Matrix for the Transfer Learning – Bidirectional LSTM Model

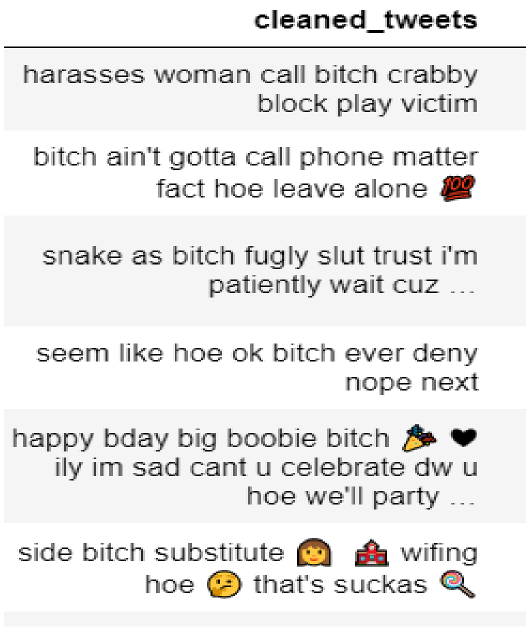
The table below is taken from the SemEval 2019 Task 5 named *HatEval* [18]. The maximum macro averaged F1-score received from the participants for hate speech detection in English (SubTask A) is 0.65 and third quantile Q3 is 0.488. Our model [macro F1-score – 0.63] was able to perform better than 75% of the participant's models. (The information related to this competition was given in the README file provided along with the datasets.). Since this project used the same data, it is more comparable with these performances.

	Subtask A		**Subtask B**	
	English	**Spanish**	**English**	**Spanish**
Min.	0.3500	0.4930	0.1590	0.4280
Q1	0.4050	0.6665	0.2790	0.5820
Mean	0.4484	0.6821	0.3223	0.6013
Median	0.4500	0.7010	0.3120	0.6160
StdDev	0.0569	0.0521	0.0890	0.0662
Q3	0.4880	0.7165	0.3570	0.6365
Max.	0.6510	0.7300	0.5700	0.7050
SVC Baseline	0.451	0.701	0.308	0.588
MFC Baseline	0.367	0.370	0.580	0.605

Table 3: Results from the SemEval 2019 entries showing the value of the best preforming model and the distribution of submission performance. This figure was created by [13].

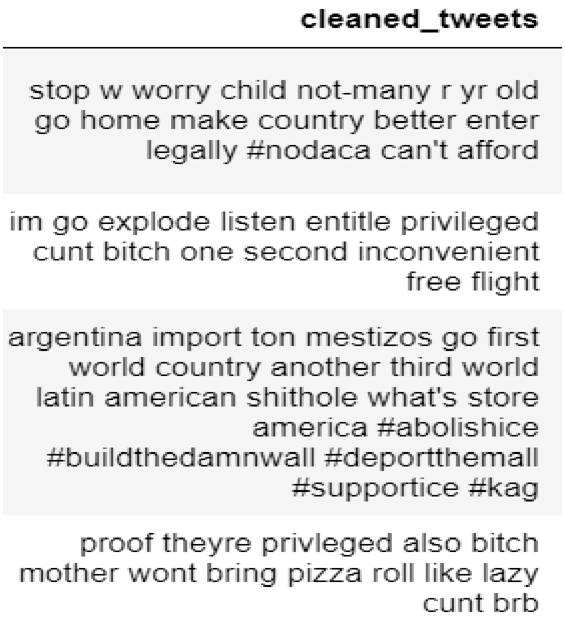
We also identified some of the false negative cases i.e., hate speech tweets predicted as Not hate speech and false positive cases i.e. Not hate speech tweets identified as hate speech.

The figure 7 and 8 shows that the model misclassified tweets having some offensive word like 'bitch' as Hate Speech though these words were used in a humorous manner in some contexts. In the case of false negatives, there are many neutral words which would have made the model to predict them as 'Not Hate Speech'.



cleaned_tweets
harasses woman call bitch crabby block play victim
bitch ain't gotta call phone matter fact hoe leave alone 💯
snake as bitch fugly slut trust i'm patiently wait cuz ...
seem like hoe ok bitch ever deny nope next
happy bday big boobie bitch 🎉 🖤 ily im sad cant u celebrate dw u hoe we'll party ...
side bitch substitute 👩 🏫 wifing hoe 🤔 that's suckas 🍭

Figure 6: Data from the testing set that had been classified as hate speech.



cleaned_tweets
stop w worry child not-many r yr old go home make country better enter legally #nodaca can't afford
im go explode listen entitle privileged cunt bitch one second inconvenient free flight
argentina import ton mestizos go first world country another third world latin american shithole what's store america #abolishice #buildthedamnwall #deportthemall #supportice #kag
proof theyre privleged also bitch mother wont bring pizza roll like lazy cunt brb

Figure 7: Data from the test set that was classified as not hate speech.

8 Conclusion and future work

In this report, we explored transfer learning with GloVe embeddings and data augmentation techniques to classify tweets as hate speech or not. Though the easy data augmentation technique proposed by Wei and Zou (2019) [14] could not perform well with this dataset, the results obtained from transfer learning approach is promising. Tuning hyperparameters for a neural network architecture is challenging and plainly relying on trial and error does not give optimal solution. Here, we explored the library – Keras Tuner provided by TensorFlow to fine-tune hyperparameters and could find significant improvement in the model performance. Our model preformed with an overall accuracy of 61% and was able to identify not hate speech tweets with more success than hate speech, despite the data not being overly biased to favour one. This models' performance is not perfect but represents a good and promising attempt at reliable hate speech detection, especially for tweets.

In future we would try to use contextual word embeddings such as ELMo or BERT and analyse their performance in hate speech detection. Another approach to explore is to use bi-directional gated recurrent unit which has become very popular in recent years in sentiment analysis. An approach similar to that used by CodaLab Competition [5] could be used, where an ensemble classifier is built, made of a variety of recurrent neural networks to improve the classification performance. This tested method has shown very effective performance with large twitter datasets.

References

[1] G. Kovács, P. Alonso, and R. Saini, 'Challenges of Hate Speech Detection in Social Media', *SN Comput. Sci.*, vol. 2, no. 2, p. 95, Feb. 2021, doi: 10.1007/s42979-021-00457-3.

[2] 'Frontiers | The Datafication of Hate: Expectations and Challenges in Automated Hate Speech Monitoring | Big Data'. https://www.frontiersin.org/articles/10.3389/fdata.2020.00003/full (accessed May 09, 2021).

[3] S. MacAvaney, H.-R. Yao, E. Yang, K. Russell, N. Goharian, and O. Frieder, 'Hate speech detection: Challenges and solutions', *PLOS ONE*, vol. 14, no. 8, p. e0221152, Aug. 2019, doi: 10.1371/journal.pone.0221152.

[4] 'The Number of tweets per day in 2020', *David Sayce*, Dec. 03, 2019. https://www.dsayce.com/social-media/tweets-day/ (accessed Apr. 27, 2021).

[5] 'CodaLab - Competition'. https://competitions.codalab.org/competitions/19935#learn_the_details-overview (accessed May 08, 2021).

[6] Y. Kim, 'Convolutional Neural Networks for Sentence Classification', in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, Oct. 2014, pp. 1746–1751, doi: 10.3115/v1/D14-1181.

[7] A. Graves and A. Mohamed, 'Speech recognition with deep recurrent neural networks | IEEE Conference Publication | IEEE Xplore'. https://ieeexplore.ieee.org/abstract/document/6638947/?casa_token=e-E-EFcIg_UAAAAA:1Bih7A820jGm9JudVXndd2mPNigp8rDGsM-ku3mfGZvFdZMBgB1-e3iqOO0_7Ll9KgxwSmH_ (accessed May 08, 2021).

[8] G. K. Pitsilis, H. Ramampiaro, and H. Langseth, ‘Effective hate-speech detection in Twitter data using recurrent neural networks’, *Appl. Intell.*, vol. 48, no. 12, pp. 4730–4742, Dec. 2018, doi: 10.1007/s10489-018-1242-y.

[9] P. Badjatiya, S. Gupta, and V. Varma, ‘Deep Learning for Hate Speech Detection in Tweets | Proceedings of the 26th International Conference on World Wide Web Companion’. https://dl.acm.org/doi/abs/10.1145/3041021.3054223?casa_token=e-M5dkmebOIAAAAA%3ABL405JYx0xWKojuz9khoeerujMzFuiJF_fO2L9_JcRyA0Sz4r6gDLiF5TR2pvsboktXi4pXhQrqX (accessed May 08, 2021).

[10] V. Indurthi, B. Syed, M. Shrivastava, N. Chakravartula, M. Gupta, and V. Varma, ‘FERMI at SemEval-2019 Task 5: Using Sentence embeddings to Identify Hate Speech Against Immigrants and Women in Twitter’, in *Proceedings of the 13th International Workshop on Semantic Evaluation*, Minneapolis, Minnesota, USA, 2019, pp. 70–74, doi: 10.18653/v1/S19-2009.

[11] E. Fehn Unsvag and B. Gambäck, ‘The Effects of User Features on Twitter Hate Speech Detection’, in *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, Brussels, Belgium, Oct. 2018, pp. 75–85, doi: 10.18653/v1/W18-5110.

[12] N. I. Pratiwi, I. Budi, and I. Alfina, ‘Hate Speech Detection on Indonesian Instagram Comments using FastText Approach’, in *2018 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, Oct. 2018, pp. 447–450, doi: 10.1109/ICACSIS.2018.8618182.

[13] A. Montejo-Ráez, S. M. Jiménez-Zafra, M. A. García-Cumbreras, and M. C. Díaz-Galiano, ‘SINAI-DL at SemEval-2019 Task 5: Recurrent networks and data augmentation by paraphrasing’, in *Proceedings of the 13th International Workshop on Semantic Evaluation*, Minneapolis, Minnesota, USA, Jun. 2019, pp. 480–483, doi: 10.18653/v1/S19-2085.

[14] J. Wei and K. Zou, ‘EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks’, *ArXiv190111196 Cs*, Aug. 2019, Accessed: May 08, 2021. [Online]. Available: http://arxiv.org/abs/1901.11196.

[15] J. Pennington, R. Socher, and C. Manning, ‘GloVe: Global Vectors for Word Representation’, in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, Oct. 2014, pp. 1532–1543, doi: 10.3115/v1/D14-1162.

[16] J. Wei, *jasonwei20/eda_nlp*. 2021.

[17] O. Levy and Y. Goldberg, ‘Dependency-Based Word Embeddings’, in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Baltimore, Maryland, Jun. 2014, pp. 302–308, doi: 10.3115/v1/P14-2050.

[18] V. Basile *et al.*, ‘SemEval-2019 Task 5: Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter’, in *Proceedings of the 13th International Workshop on Semantic Evaluation*, Minneapolis, Minnesota, USA, Jun. 2019, pp. 54–63, doi: 10.18653/v1/S19-2007.