TOXIC COMMENT CLASSIFICATION COMPARISON BETWEEN LSTM, BILSTM, GRU, AND BIGRU

[1]Jonathan Kevin Giustino, [2]Yonathan Purbo Santosa
[1,2]Program Studi Teknik Informatika Fakultas Ilmu Komputer,
Universitas Katolik Soegijapranata
[2]yonathansantosa@unika.ac.id

ABSTRACT

One of the biggest problems in modern internet age is toxicity. In this study, our aim is to draw an effective method to classify toxicity in text comment in this case specifically on Wikipedia comment. Old Regular Rule based model attacks, Machine Learning Methods suffer from rules based approaches and thus are incapable of accurate detection of toxicity while maintaining precision at the same time. To overcome this limitation, recurrent neural networks (RNNs), wherein, long short-term memory (LSTM) networks, gated recurrent unit(GRU) have been proposed. In this paper, author compares the LSTM, BiLSTM, GRU, and BiGRU for multi label classification for which is the best model to use towards jigsaw toxicity challenge dataset to classify toxicity. This study will be finding out which of the model is the best for classification and the difference between different type of pre-processing. We'll go ahead to use League of Legends tribunal datasets from kaggle as our base. The results obtained were that, the highest accuracy on the test was attained by BiLSTM model without cleaning with on 87.208% accuracy, 55.205% Precision, 68.540% Recall, and 60.623% F1-Score the result also shows while preprocessing on cleaning improve the resultant metrics by a marginal amount for regular LSTM, GRU, and BiGRU it doesn't always improve the result.

Keywords: Toxicity, LSTM, GRU, BiLSTM, BiGRU

INTRODUCTION

The incidence of a toxic and abusive display within the online has been on the rise with harmful effects commensurate to the persons as well as communities. These behaviors may manifest as hate speeches, cyberbullying, or harassment which eventually ends up stressing, making others underachieve, or even causing suicidal thoughts among victims [1]. This calls for the professional identification and eradication of toxic language with an effort to create a safe online environment.

Some of the approaches in the past that have been adopted in toxicity detection include rule-based models, machine learning, deep learning among others. However, these approaches are characterized by a low level of accuracy and lack scalability as well as no capability of the understanding of nuanced language expressions [2]. Basic rule systems cannot capture nuanced meanings and machine / deep learning models require humongous amount of pre-labeled datasets which may not be always abundant for various domains or languages [2].

Recent studies have shown the effectiveness of Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks in extracting information from complex, long-range sources. However, deep learning models are efficient models in handling sequential data such as text inputs by capturing dependencies over a long period [3].

The general technique Word Embedding with the recurrent neural networks based on LSTM helps to establish semantic relations among words that set the premise for natural language processing. The above techniques intend to be worked out in this paper with LSTM, Bidirectional LSTM, GRU and Bidirectional GRU in order to classify toxicity in online socialization [4]. A significant challenge is identifying and implementing previous models used in the classification of comments based on a set of rules. The focus is improving from past rule-based models and other programming learning mechanisms. Categorically, the results of the classification will outline the types such as Toxic, Severe Toxic, Obscene, Threat, Insult, and Identity Hate.

LITERATURE STUDY

Several studies have delved into the intricate realm of online toxicity detection, employing diverse methodologies and datasets. In [1], Murnion et al. made a large-scale analysis of text on the dataset of the online multiplayer game World of Tanks. Despite using sentiment analysis points in the Microsoft Azure Cognitive Service and Twin Word Sentiment Analysis, even their results turned out to be unexpectedly poor. The study adds two possible reasons, limited technological familiarity and the specialty of in-game communication as a reason for lack of optimum utilization of the services by means of short code words or phrases rather than fully written sentences.

Agreeing with Murnion, Marten et al. [2] researched on the language intricacy in Dota 2 text chat data during a match. Traditional natural language processing techniques worked ineffectively due to elliptical expressions and grammatical variations as subject-oriented irregularity of the language. To mitigate such challenges, the research used specialized tokenization and n-grams correlating toxicity with game outcomes through SVM and TF-IDF.

Dubey et al. [3] focused on the efficiency of Long Short-Term Memory (LSTM) networks in comment classification as toxic. Showing exemplary accuracy, precision, and recall, the LSTM model provided a sense of feel of the severity of toxic in sentences.

Esposito et al. [5] highlighted a critical problem in models trained with imbalanced data. The authors offered an approach to optimize the class-specific classification thresholds to improve the performance of the model, especially in cases when the imbalanced data may cause over-prediction of the majority class.

Ibrahim et al. [6] sought to improve the classification of toxic comments through an optimal model that used Convolutional Neural Networks (CNNs), bidirectional Long Short-Term Memory (LSTM) as well as bidirectional Gated Recurrent Units (GRUs). Its ensemble model outperformed

the prior technique with having high F1 score for both toxic/non-toxic categorization and predicting toxicity categories.

In following the methodology used by Ibrahim, Anand and Eswar [4], classification of toxic categories in Bengali social media remarks was considered. They have used LSTM, Artificial Neural Networks (ANN), besides comparisons between them as well as further assessments consisting of CNN, GloVe & LSTM, and a combination of LSTM and ANN.

Mossie and Wang [7] further extended the toxic detection to low-resource languages but focused on Amharic. They registered that Word2Vec embedding and RNN-GRU worked best in identifying hate speech with high area under the curve (AUC) and accuracy.

Obadimu et al. [8] used the Perspective API which is a CNN-based classifier to assess the felt toxicity in YouTube comments. A further stratum of unsupervised analysis by using Latent Dirichlet Allocation (LDA) topic modeling brought out semantic patterns operating beneath the unstructured text bodies.

Abbasi et al. [9] pointed out a specific challenge in toxicity classification models—sensitivity to groups of identities often targeted. Their RNN model showed better performance on Kaggle datasets with an emphasis on precision, recall, and F1-score.

Insights synthesized by Yazgili and Baykara [10] from different sources highlighted that the sentence analysis approach should focus more on the purposeful analysis as compared to being just a word-based one. Highlighting context and intent of sentences, to be taken into considering effectiveness in detecting toxicity, were indicated to be important.

From an unusual angle, Mohamad [11] pursued the impact of pre-processing techniques on state-of-the-art models. Quite in contrast to general conventions, at least one work proposed that with minimal transformations, one can obtain good performance which is effective in warning against over-preprocessing measures that are self-defeating as well.

RESEARCH METHODOLOGY

The research will follow flowchart, displayed on Figure 3.1. First, we prepare and preprocess our dataset. We need to make sure our dataset can be processed by both our GRU and LSTM. These results will be evaluated and analyzed, in search for conclusive result and creating a comprehensive report.

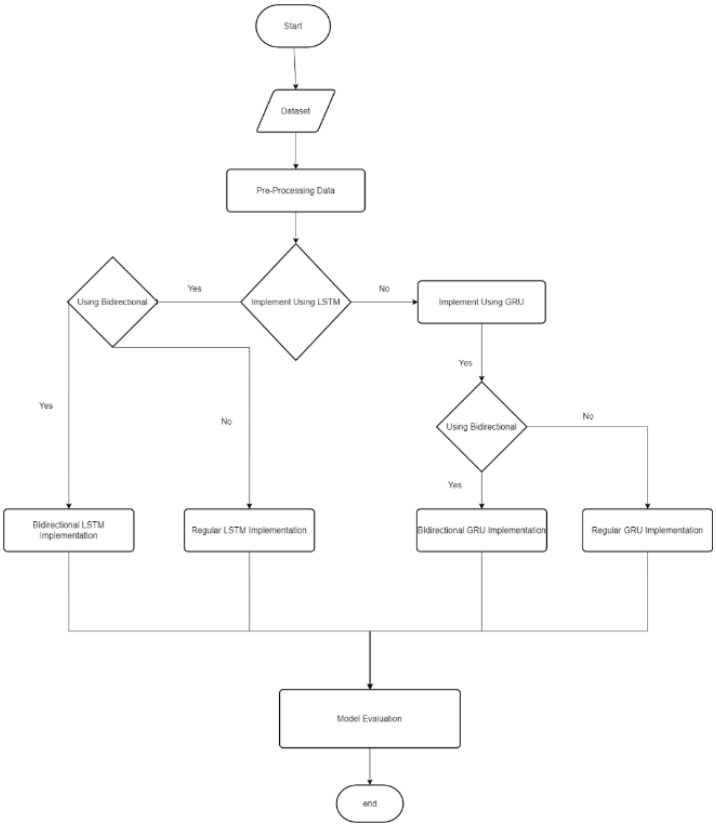

Figure 1. Research Methodology

Dataset Collection

The study will use jigsaw-toxic-comment-classification-challenge dataset available at kaggle [12]. This dataset consists of text chat of League of Legends reported cases before it went down. This dataset consist of 9370 data that include shown in Figure 3.2:

1. **train.csv** - the training set, contains comments with their assigned binary labels
2. **test.csv** - the test set which will be used for our testing
3. **sample_submission.csv** - a sample submission file to submit in kaggle competition
4. **test_labels.csv** – real labels for the test data

Dataset contains 6 toxic behavior labeled by human raters. The types of toxicity are:

- toxic
- severe_toxic
- obscene
- threat
- insult

• identity_hate

Data Pre-processing

We arranged pre-processing tasks with combination of Murnion et al method [1] and Marten [2] method. We use 3 different types of preprocessing on our train data that will be each implemented on our model are:

• Cleaned : Removed non aplhanumeric, unnecessary character, and stopword removal

• Cleaned without using stopword removal : Removed non aplhanumeric and unnecessary character

• Not Cleaned : Plain training dataset

After being pre-processed the train data will be splitted into 80% used for training data and 20% used for validation and optimizing purpose. Testing will use test.csv provided on the dataset

Model Development

Our multi-label classifier model will be implemented using deep learning frameworks, TensorFlow. Taking into account such crucial components as input dataset format, total layers augmented implementations, strong activation implementations and overall concordant harmonic reverberation is paramount. There will be 4 model created for this research. Model with LSTM, model created with Bidirectional LSTM, model created with GRU and model created with Bidirectional GRU. All model well be tuned by their hyper parameter, We'll tune out the epoch size, batch size, layer size, and threshold to find out the best result.

LSTM Algorithm

Long Short-Term Memory (LSTM) is one of RNN (Recurrent Neural Network) model designed to capture long-term dependencies in sequential data. LSTM have memory cells and three gates (input, forget, and output), this method is able to address vanishing gradient problem on regular RNN. LSTMs usually used on natural language processing task due to the ability to retain and selectively use information over extended sequences.

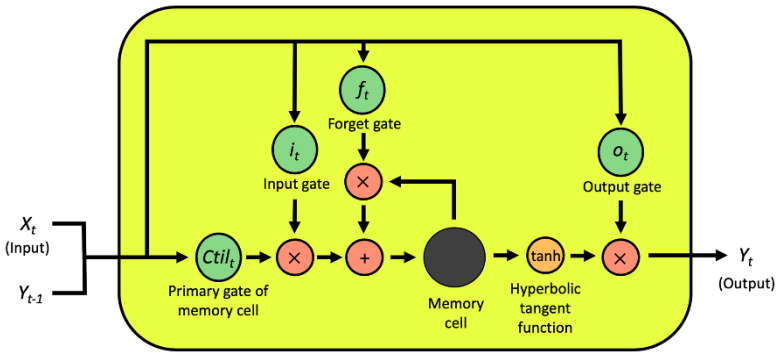

Figure 2. LSTM Model

Bidirectional LSTM Algorithm

Bidirectional Long Short-Term Memory (BiLSTM) is a version of LSTM designed to capture information from two direction, past and future in sequential data hence the bidirectional term. Unlike regular LSTM, in BiLSTM the input flows in both directions and can utilize information from both sides. At each step of the sequence, information flows backward in time from both the past and future through their gated hidden states at every time step, thereby making the model more context-aware and thus considerably have better performance for regular LSTM in most case scenario with the tradeoff being slower time to train.

GRU Algorithm

Gated Recurrent Unit (GRU) is a type of RNN with similar architecture to LSTM and also designed to mitigate vanishing gradient problem in sequential data. Instead of LSTM three main gate, GRU simplifies the structure by only having two gates: reset gate and update gate, resulting in lighter. While being usually used for similar purpose which is natural language processing, GRU is computationally less intensive and more fitting for the tasks with lower data count compared with LSTMs.

Bidirectional GRU Algorithm

Bidirectional Gated Recurrent Unit (BiGRU) is a development of GRU which both collects information in two directions – the past and the future – with respect to the current time step in the input sequence. Just like BiLSTM, BiGRU allows capturing dependencies between backward and forward ones on each other. thereby making the model more context-aware and thus considerably have better performance compared with regular GRU with the tradeoff being slower time to train.

Model Evaluation and Analysis

In this research, the analysis will evaluate and compare the performance of LSTM, BiLSTM, GRU, and BiGRU. The author will record the result of accuracy, precision, recall and f1-score of each model and the effect of preprocessing applied.

After all results have been recorded comparing the result between each model, the author will be made to be able to see the comparative value between the four algorithm models. By comparing all algorithm models, we can understand which model is better and more efficient in classifying toxicity.

IMPLEMENTATION AND RESULT

The author planned to make true or false prediction for each label instead of sigmoid number between 0 and 1 so we need to find threshold [5], to find threshold author used validation dataset and threshold from 0,1 to 0,9 with 0,1 increment and finding precision median of each model, the result will be used for all model with testing data.

Table 1. Threshold Found on all models

Method	Threshold

LSTM cleaned + stopword	0,4
LSTM cleaned no stopword	0,4
LSTM not cleaned	0,3
Bidir LSTM cleaned + stopword	0,5
Bidir LSTM cleaned no stopword	0,5
Bidir LSTM not cleaned	0,5
GRU Cleaned + stopword	0,5
GRU Cleaned no stopword	0,5
GRU not cleaned	0,5
BiGRU Cleaned + stopword	0,5
BiGRU Cleaned no stopword	0,5
BiGRU not cleaned	0,5

After finding the threshold on validation data for each of the model with their own preprocessing, the threshold is then implemented into test data to show which one is the best at multi label classification for toxic comment.

Table 2. Model Result on test data

Implement Test Data					
Method	**Threshold**	**Accuracy**	**Precision**	**Recall**	**F1-Score**
LSTM cleaned + stopword	0,4	84,493%	43,459%	72,700%	54,177%
LSTM cleaned no stopword	0,4	86,005%	49,131%	70,796%	57,773%
LSTM not cleaned	0,3	85,017%	44,989%	73,934%	55,851%
Bidir LSTM cleaned + stopword	0,5	86,674%	54,814%	68,747%	60,128%
Bidir LSTM cleaned no stopword	0,5	86,083%	50,554%	70,955%	58,664%
Bidir LSTM not cleaned	0,5	87,208%	55,205%	68,540%	60,623%
GRU Cleaned + stopword	0,5	81,848%	44,922%	75,045%	54,249%
GRU Cleaned no stopword	0,5	84,917%	50,144%	72,106%	57,874%
GRU not cleaned	0,5	84,695%	45,934%	74,452%	55,970%
BiGRU Cleaned + stopword	0,5	77,927%	45,097%	77,149%	53,492%
BiGRU Cleaned no stopword	0,5	85,931%	49,748%	74,652%	59,535%
BiGRU not cleaned	0,5	85,681%	52,854%	71,231%	59,960%

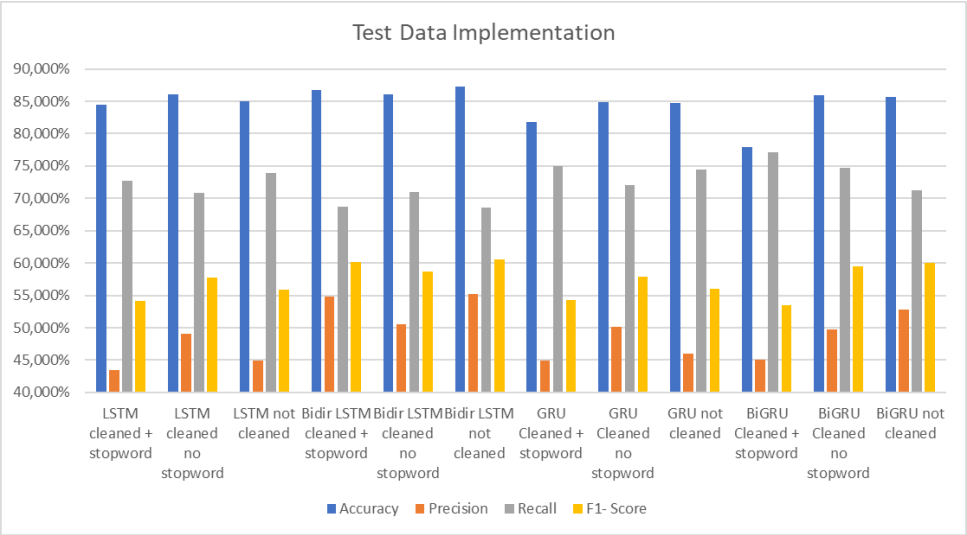

Figure 3. Result on test data Implementation

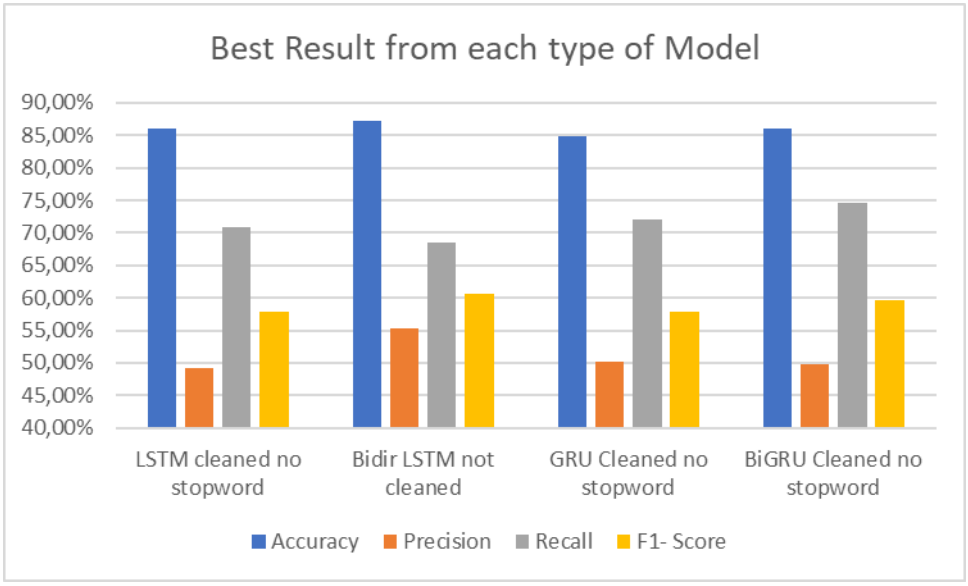

Figure 4. Best Accuracy Result of each model

The result found on table 2 shown on both figure 3 and 4, the author found that BiLSTM model is the best performing model compared to the other three model by having the highest accuracy value, precision value and f1-score compared to regular LSTM and both of GRU Model

Table 3. BiLSTM Result on test data

Implement Test Data					
Method	**Threshold**	**Accuracy**	**Precision**	**Recall**	**F1- Score**
Bidir LSTM cleaned + stopword	0,5	86,674%	54,814%	68,747%	60,128%
Bidir LSTM cleaned no stopword	0,5	86,083%	50,554%	70,955%	58,664%
Bidir LSTM not cleaned	0,5	87,208%	55,205%	68,540%	60,623%

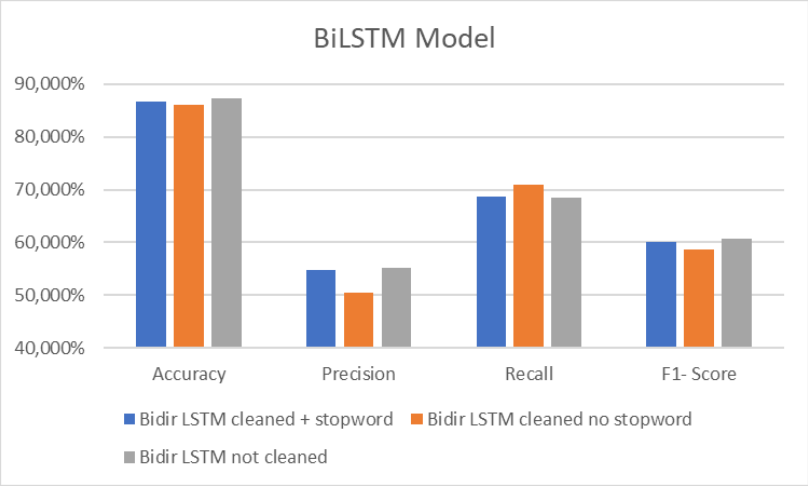

Figure 5. BiLSTM result on test data

On BiLSTM model shown at table 3 and figure 5, cleaned + stopword impact is slightly worse than non-cleaned one that is close within 0.5 between accuracy of 86,674% in cleaned + stopword and 87,208% in non-cleaned one , and on f1-score 60,623% in cleaned + stopword and 60,128 in non-cleaned. while prior model result is close cleaned without stopword yielded worse result compared to the other two in all except recall value.

Table 4. GRU and BiGRU Result on test data

Implement Test Data					
Method	**Threshold**	**Accuracy**	**Precision**	**Recall**	**F1- Score**
GRU Cleaned + stopword	0,5	81,848%	44,922%	75,045%	54,249%
GRU Cleaned no stopword	0,5	84,917%	50,144%	72,106%	57,874%
GRU not cleaned	0,5	84,695%	45,934%	74,452%	55,970%
BiGRU Cleaned + stopword	0,5	77,927%	45,097%	77,149%	53,492%
BiGRU Cleaned no stopword	0,5	85,931%	49,748%	74,652%	59,535%
BiGRU not cleaned	0,5	85,681%	52,854%	71,231%	59,960%

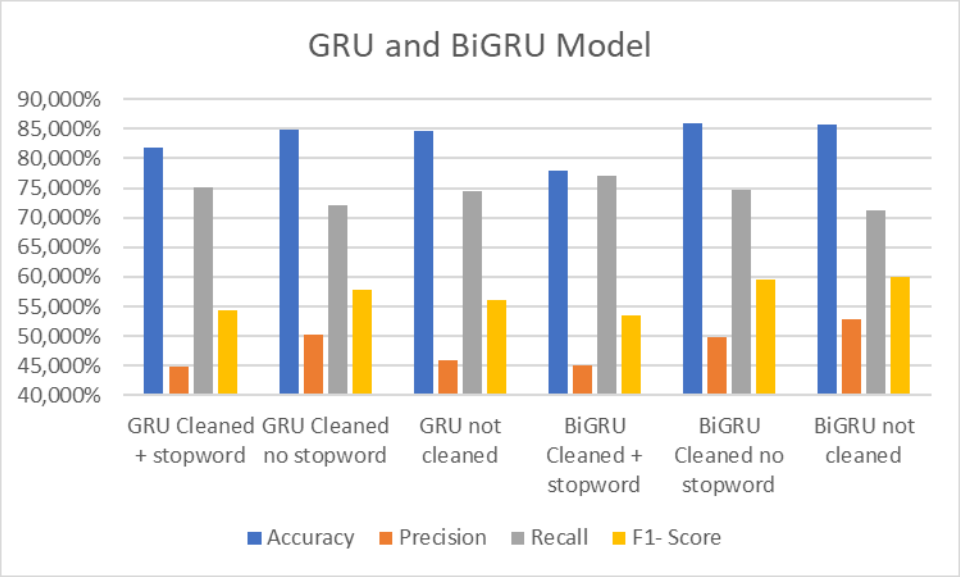

Figure 6. GRU and BiGRU result on test data

Shown at table 4 and figure 6, the result of preprocessing on GRU and BiGRU model, using cleaned train data and no stopword removal provide increase in accuracy, and f1-score, and in regular GRU case is also increasing the precision value by up to 5% from 45,934% to 50,144%. The same method is reducing precision value n BiGRU model with precision from 42.while cleaned + stopword yet again having worst result in all category except for recall.

Table 5. Regular LSTM Result on test data

Implement Test Data					
Method	**Threshold**	**Accuracy**	**Precision**	**Recall**	**F1- Score**
LSTM cleaned + stopword	0,4	84,493%	43,459%	72,700%	54,177%
LSTM cleaned no stopword	0,4	86,005%	49,131%	70,796%	57,773%
LSTM not cleaned	0,3	85,017%	44,989%	73,934%	55,851%

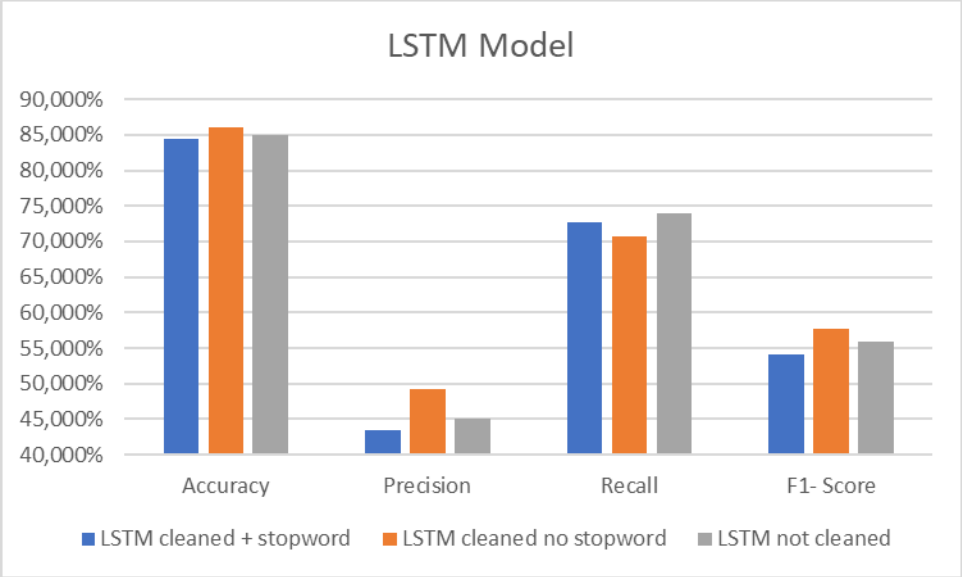

Figure 7. GRU and BiGRU result on test data

On regular LSTM shown at table 5 and figure 7, cleaned without stopword removal method yielded clear improvement in all category except recall, with non-cleaned method following second on both precision and accuracy. With all of the result in mind even though cleaning data can improve some in some metrics, not cleaning the data can provide with consistent base line across all models. This might be a result of higher relevant information that might be missing as a result of cleaning process.

On all of the model they provide relatively high accuracy but model precision and F1-score value are lower than desired, this is indicating that all models are making a lot of false positive prediction that could be problematic case scenario on toxicity classification meaning non toxic user can be flagged for toxicity. This is most likely this case of the dataset being heavily imbalanced. To mitigate this, we can implement better classification threshold or do a better job at feature engineering our result can most likely be improved upon.

CONCLUSION

In this research, the author compares LSTM, BiLSTM, GRU, and BiGRU on toxicity classification. From the result it can be concluded that all of the models can be used for toxicity classification. but on the test data BiLSTM model outperform the other model. The best resulting BiLSTM is the one that is not applied with any cleaning with 87,208% Accuracy, 55,205% Precision, 68,540% Recall, and 60,623% F1-Score followed by BiLSTM model with cleaning and stop word removal with the result 86,674% Accuracy, 54,814% Precision, 68,747% Recall, and 60,128% F1-score.

Preprocessing with cleaning the data without removing the stopwords improved results metric in accuracy, precision, and f1-score on LSTM, GRU, BiGRU. This improvement while noticeable, is still relatively close to base result without any kind of cleaning. The author believes it is more impactful to the results, by modifying the model structure itself to suit the dataset better will result in better performance metrics by avoiding underfitting/overfitting rather than focusing on different type of preprocessing.

All models have relatively high accuracy but sacrifice in precision value and F1-score value, this is indicating that all models is making a lot of false positive prediction that could be problematic case scenario on toxicity classification meaning nontoxic user can be flagged for toxicity. This is most likely this case of the dataset being heavily imbalanced. By implementing better classification threshold or do a better job at feature engineering especially specifying the case for multi label classification result metrics can most likely be improved upon.

For the suggestion, author believes that model based on this paper can be developed more by adding different type of preprocessing, training and implementing the model on dataset with different languages, and modifying model structure further to suit the dataset. Author also recommends to trying out many kinds of pre trained embedding instead of using custom embedding such as Word2Vec, BERT, GloVe, and other to further find the impact of each type of embedding in toxicity classification.

DAFTAR PUSTAKA

[1] S. Murnion, W. J. Buchanan, A. Smales, and G. Russell, “Machine learning and semantic analysis of in-game chat for cyberbullying,” *Computers & Security*, vol. 76, pp. 197–213, Jul. 2018, doi: 10.1016/j.cose.2018.02.016.

[2] M. Martens, S. Shen, A. Iosup, and F. Kuipers, “Toxicity detection in multiplayer online games,” in *2015 International Workshop on Network and Systems Support for Games (NetGames)*, Zagreb: IEEE, Dec. 2015, pp. 1–6. doi: 10.1109/NetGames.2015.7382991.

[3] K. Dubey, R. Nair, Mohd. U. Khan, and Prof. S. Shaikh, “Toxic Comment Detection using LSTM,” in *2020 Third International Conference on Advances in Electronics, Computers and Communications (ICAECC)*, Bengaluru, India: IEEE, Dec. 2020, pp. 1–8. doi: 10.1109/ICAECC50550.2020.9339521.
[4] M. Anand and R. Eswari, “Classification of Abusive Comments in Social Media using Deep Learning,” in *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, Erode, India: IEEE, Mar. 2019, pp. 974–977. doi: 10.1109/ICCMC.2019.8819734.
[5] C. Esposito, G. A. Landrum, N. Schneider, N. Stiefl, and S. Riniker, “GHOST: Adjusting the Decision Threshold to Handle Imbalanced Data in Machine Learning,” *J. Chem. Inf. Model.*, vol. 61, no. 6, pp. 2623–2640, Jun. 2021, doi: 10.1021/acs.jcim.1c00160.
[6] M. Ibrahim, M. Torki, and N. El-Makky, “Imbalanced Toxic Comments Classification Using Data Augmentation and Deep Learning,” in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Orlando, FL: IEEE, Dec. 2018, pp. 875–878. doi: 10.1109/ICMLA.2018.00141.
[7] Z. Mossie and J.-H. Wang, “Vulnerable community identification using hate speech detection on social media,” *Information Processing & Management*, vol. 57, no. 3, p. 102087, May 2020, doi: 10.1016/j.ipm.2019.102087.
[8] A. Obadimu, E. Mead, M. N. Hussain, and N. Agarwal, “Identifying Toxicity Within YouTube Video Comment,” in *Social, Cultural, and Behavioral Modeling*, vol. 11549, R. Thomson, H. Bisgin, C. Dancy, and A. Hyder, Eds., in Lecture Notes in Computer Science, vol. 11549. , Cham: Springer International Publishing, 2019, pp. 214–223. doi: 10.1007/978-3-030-21741-9_22.
[9] A. Abbasi, A. R. Javed, F. Iqbal, N. Kryvinska, and Z. Jalil, “Deep learning for religious and continent-based toxic content detection and classification,” *Sci Rep*, vol. 12, no. 1, p. 17478, Oct. 2022, doi: 10.1038/s41598-022-22523-3.
[10] E. Yazgili and M. Baykara, “Cyberbullying and Detection Methods,” in *2019 1st International Informatics and Software Engineering Conference (UBMYK)*, Ankara, Turkey: IEEE, Nov. 2019, pp. 1–5. doi: 10.1109/UBMYK48245.2019.8965514.
[11] F. Mohammad, “Is preprocessing of text really worth your time for online comment classification?” arXiv, Aug. 29, 2018. doi: 10.48550/arXiv.1806.02908.
[12] “jigsaw-toxic-comment-classification-challenge.” Accessed: Jan. 05, 2024. [Online]. Available: https://www.kaggle.com/datasets/julian3833/jigsaw-toxic-comment-classification-challenge