JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 3, September 2023, Pp. 764-773

DETECTION OF INDONESIAN HATE SPEECH IN THE COMMENTS COLUMN OF INDONE-SIAN ARTISTS' INSTAGRAM USING THE ROBERTA METHOD

Adhe Akram Azhari[1], Yuliant Sibaroni[2], Sri Suryani Prasetiyowati[3]

1. Informatics, School Of Computing, Telkom University, Bandung, Indonesia
2. Informatics, School Of Computing, Telkom University, Bandung, Indonesia
3. Informatics, School Of Computing, Telkom University, Bandung, Indonesia

Article Info

Keywords: Instagram; RoBERTa; Full Pre-Processing; Non Full-PreProcessing; Hate Speech

Article history:
Received 6 March 2023
Revised 20 March 2023
Accepted 3 April 2023
Available online 1 September 2023

DOI :
https://doi.org/10.29100/jipi.v8i3.3898

* Corresponding author.
Adhe Akram Azhari
E-mail address:
adheakramazhari@students.telkomuniversity.ac.id

ABSTRACT

This study detects hate speech comments from Instagram post comments where the method used is RoBERTa. Roberta's model was chosen based on the consideration that this model has a high level of accuracy in classifying text in English compared to other models, and possibly has good potential in detecting Indonesian as used in this research. There are two test scenarios namely full-preprocessing and non full-preprocessing where the experimental results show that non full-preprocessing has an average value of accuracy higher than full-preprocessing, and the average value of non full-preprocessing accuracy is 85.09%. Full-preprocessing includes several preprocessing stages, namely cleansing, case folding, normalization, tokenization, and stemming. While non full-preprocessing includes all processes in preprocessing except the stemming process. This shows that RoBERTa predicts comments well when not using full-preprocessing.

I. INTRODUCTION

HATE speech is an act that shows hatred towards one person or group. This action aims to have a certain impact either directly or indirectly [1]. Most Indonesian people take this action through social media. Because social media provides a comment feature where everyone is free to give their opinion but not a few people misuse this feature. The impact of this action is in the form of violence, depression, violence, and social conflict [1]. For example, Indonesian artists often get hate speech from the public. This action can make the artist become depressed and not rule out the possibility of taking his life. Therefore this action must be watched out for.

Instagram is one of the social media that is often used by the community, especially the people of Indonesia today. It has been noted that Instagram users are growing faster than Facebook. Instagram users have increased by 4.5 percent since the start of 2020[2]. This indicates that there are more Instagram users than on Facebook. There are several features provided by Instagram to upload photos, and videos and make comments. Users can take advantage of the comments feature to give their opinions regarding photos or videos that have been uploaded. Not a few Instagram users abuse the function of the comments feature. For example, spreading hate speech through the Instagram comment column to the person who uploaded the photo or video.

In this increasingly rapid technological development, many Indonesian people use the internet to carry out their daily activities. Especially in 2020, Indonesia was affected by COVID-19 where people were forced to do work at home. According to research conducted by We Are Social on April 23, 2020, Instagram users are growing faster than Facebook[2]. Instagram recorded an increase of 4.5 percent since the beginning of 2020. Globally Instagram is ranked fifth as a social media that has active users [2].

Artists in Indonesia mostly use Instagram in carrying out their work such as doing product promotions and uploading their works. Of course, after uploading content there will be a comment feature where people are given the opportunity to give praise, suggestions, and input to the artist. But not a few people blame this feature by giving comments in the form of hate. Most people feel free to give comments so they can be abused. Therefore there are often hate comments. Freedom in terms of commenting makes most Indonesian people not afraid to comment on their social media accounts.

So that Indonesia formed a Virtual Police program that aims to find social media accounts that provide comments in the form of hate speech[3]. In 2021 the Virtual Police program has been formed and has found 415 social media accounts that often display hate speech[3]. Comments are often found under the guise of criticism but these comments contain hate speech[3].

Based on the problems that occur, there are many studies that take the topic of detecting hate speech. To support the formation of a hate speech detection tool, machine learning is needed. Many studies use machine learning in detecting hate speech. In 2018 a hate speech detection machine was made using the CNN method[5]. This study obtained an average accuracy of 99.8%, a precision of 99.46%, and a recall of 97.99%[5]. This research get 2 types of output, namely comments that include hate speech or non-hate speech. In previous research, CNN serves to form patterns that will present classifications, and also previous researchers hoped that by using the Deep Learning method with the CNN algorithm, it would be able to detect an image that contains elements of hate speech. In 2021, research will be conducted on social media Twitter to detect hate speech [6]. This study uses a support vector machine to perform classification. In this study compared three kernels namely the RBF, linear, and sigmoid kernels. The results of this study RBF kernel has the highest accuracy value.

Next is research conducted by Yunita Suryani, et al. who once conducted a study related to hate speech on the Instagram social media of one Indonesian artist in 2021[7]. This study uses descriptive and qualitative methods [7]. This study aims to describe hate speech made by people who hate the artist. In 2020 a study was conducted regarding the analysis of comment sentiment on Instagram social media[8]. This study used the TF-IDF method and the naïve Bayes classifier in carrying out the classification. The results of this study are that this method can detect hate speech on Instagram social media by 92%[8].In 2019 Sakti Putra Perdana B.B once made a hate speech detection machine in the Instagram comment column[9]. This study used a deep neural network classification method. The results of this study are the average precision, recall, and F1 values of 97% and 97.19% accuracy and an average classification time of 5.22 seconds[9]. Annisa Briliani has also made a hate speech detection machine in the Instagram comment column using the k-nearest neighbor classification method [10]. The results of this study showed an average value of precision, recall, and F1 of 96% and 96.22% accuracy [10]. In 2019 Elvira Erizal also made a machine for detecting hate speech in the Instagram comment column using the maximum entropy classification method[11]. The results of this study obtained a precision value of 92.18%, a recall of 90.38%, an F1 of 90.44%, and an accuracy of 90.56% [11].

Yinhan Liu, et al conducted a study related to RoBERTa in 2019 [12].There are, 160GB of data was used to be examined. This study uses three different benchmarks to measure the performance of the RoBERTa model, namely GLUE results, SQuAD results, and RACE results. In the GLUE results, the performance of RoBERTa is superior compared to the large XLNET and large BERT models [12]. Furthermore, the results of SQuAD results RoBERTa get the highest accuracy value compared to XLNET large and BERT large [12]. In the last benchmark, RoBERTa's RACE results again outperformed the other models because RoBERTa's accuracy reached 86.5% [12].

In 2021 there will be research related to the uses of the RoBERTa method in detecting English [4]. In this study to compare the accuracy of several machine learning in detecting English, there are SVM, Logistic Regression, RoBERTa, and Meta-Classifier [4]. Of the four models, Roberta has the highest level of accuracy, namely 0.695 [4]. This proves that RobertTa is able to detect language more accurately than the fourth machine learning that has been compared [4]. Based on the latest research in 2021 proves the accuracy level of RoBERTa is higher compared to other models in detecting English. This shows that RoBERTa excels in detecting the English language. So this research will conduct trials on the RoBERTa method in detecting Indonesian.

This study detects hate speech comments in Indonesian. This study uses the RoBERTa method because the RoBERTa method has a high accuracy value in detecting language compared to other machine learning. This study aims to determine whether the use of full preprocessing can affect the accuracy level of the Roberta model in detecting hate speech in Indonesian and in general to see whether the Roberta model is good enough in detecting hate speech in Indonesian language social media.

II. Research Method

This research builds a system design that aims to facilitate the detection of hate speech in the Instagram comment column. There are several stages that must be done to get optimal results, namely data collection on the Instagram comment column. The source of the dataset is taken from 3 Indonesian artists who have a lot of hate speech in the comments column of their Instagram posts. Data pre-processing is normalizing the dataset that has been collected. The data splitting is dividing the dataset into train data and test data. Next is the pre-training model from the results of the RoBERTa model training. After the model is formed, a comparison of the accuracy values between full-PreProcessing and non-full-PreProcessing is performed., besides that, it can also use a confusion matrix to see the performance of the RoBERTa model.

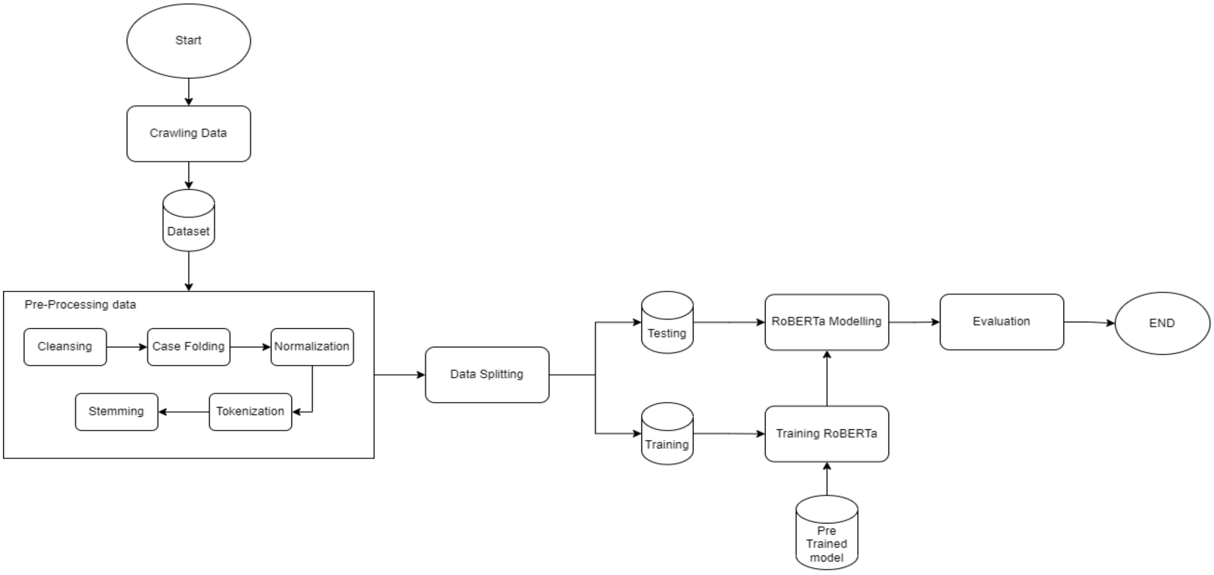

Fig. 1. Design System

Fig 1 is the design of the Indonesian hate speech detection system in the Instagram comment column. The flow of system design through several stages. The first stage is to collect the dataset from the comments column of three Instagram artists, after the dataset is collected, the pre-processing stage will be carried out on the dataset. After the dataset has passed the pre-processing stage, it enters the data splitting stage where the dataset is divided into two, namely testing and training. After being divided into two, the next stage is the RoBERTa modeling stage and the Pre-Trained model. And the last stage is evaluation.

A. Data Crawling

In the process of data crawling or collecting data for processing, there are several stages that will be carried out as follows:

1) Choose 3 Indonesian artists who most often get comments in the form of hate speech.
2) Crawling data in the comments column for 3 Indonesian artist Instagram accounts that most often get comments in the form of hate speech.
3) The results of crawling data will be stored in CSV/xlsx files. The data file will be stored in 3 files consisting of the artist's name and comments. The file corresponds to 3 artists who often get comments in the form of hate speech.

B. Pre-Processing

In the data preprocessing stage, several processing steps will be carried out to obtain more structured data so that the data is easy to process as research. The following are the stages involved in data preprocessing:

1) Cleansing Data

This stage is the first stage in pre-processing. In this stage all symbols, punctuation marks and others will be removed, so that after cleansing there will only be a sentence without punctuation marks, symbols, and others[14].

TABLE I
CLEANSING DATA

Before Cleansing	After Cleansing
BEDA AG4M4 NTR JUGA GA JODOH😂😂😂😂 MALU AMA JILBAB, KADANG KOSTUM ITU MENIPU..	BEDA AGM NTR JUGA GA JODOH MALU AMA JILBAB KADANG KOSTUM ITU MENIPU

Table 1 is the initial stage of pre-processing. Table 1 shows a comment before cleansing and after cleansing. The output that can be seen is that symbols, numbers, and punctuation marks disappear after cleansing.

2) Case Folding

This stage is the second stage in pre-processing. In this stage the data or comments that have gone through the cleansing stage will be case folding, namely changing uppercase letters to lowercase letters, so that the output that will appear is a comment with all letters being small[[14].

TABLE 2
CASE FOLDING

Before Case Folding	After Case Folding
BEDA AGM NTR JUGA GA JODOH MALU AMA JILBAB KADANG KOSTUM ITU MENIPU	beda agm ntr juga ga jodoh malu ama jilbab kadang kostum itu menipu

Table 2 is the stage after cleaning the comments. Table 2 shows the comments before case folding and after case folding. You can see the output where all letters are lowercase.

3) Normalization

This stage is the third stage of pre-processing. At this stage comments that have incomplete or unclear words will be changed into clearer words, such as "agm" will be changed to "agama" and also "ntr" will be changed to "nanti".

TABLE 3
NORMALIZATION

Before Normalization	After Normalization
beda agm ntr juga ga jodoh malu ama jilbab kadang kostum itu menipu	beda agama nanti juga tidak jodoh malu sama jilbab kadang kostum itu menipu

Table 3 is the stage after the case folding process is carried out. In this table comments that have unclear or short words will be clarified after normalization. Can be seen the difference between before and after normalization.

4) Tokenization

This stage is the fourth stage in pre-processing. At this stage, all words will be broken down into just one word. This tokenization stage will help the RoBERTa model in determining the accuracy value[[14].

TABLE 4
TOKENIZATION

Before Tokenization	After Tokenization
beda agama nanti juga tidak jodoh malu sama jilbab kadang kostum itu menipu	['beda', 'agama', 'nanti', 'juga', 'tidak', 'jodoh', 'malu', 'sama', 'jilbab', 'kadang', 'kostum', 'itu', 'menipu']

Table 4 is the comment tokenization table, where comments that have been normalized will be broken. It can be seen clearly in table 4 the difference between before and after tokenization.

5) Stemming

This is the final stage of pre-processing. At this stage, all words that have affixes will be removed, such as "deceive" is changed to "deceive"[14].

TABLE 5
STEMMING

Before Stemming	After Stemming
['beda', 'agama', 'nanti', 'juga', 'tidak', 'jodoh', 'malu', 'sama', 'jilbab', 'kadang', 'kostum', 'itu', 'menipu']	['beda', 'agama', 'nanti', 'juga', 'tidak', 'jodoh', 'malu', 'sama', 'jilbab', 'kadang', 'kostum', 'itu', 'tipu']

Table 5 is the final process of pre-processing. Table 5 shows before and after stemming. The output from table 5 is that words that previously had affixes will be deleted and become not using affixes after going through the stemming process.

C. RoBERTa Modelling

RoBERTa (A Robustly Optimized BERT Pretraining Approach) is the result of a modification of BERT. RoBERTa is trained using dynamic masking, full sentences without NSP loss, large mini-batches, and a larger byte-level BPE[12]. In the RoBERTa method, there is masking, this aims to avoid the same mask. Previous studies have compared static masking and dynamic masking. The results of this comparison show that dynamic masking can accommodate more data [12]. In the RoBERTa method, there are full sentences without NSP loss where the goal is to delete sentences that are connected to each other because these sentences will increase downstream task performance [12]. RoBERTa uses large mini-batches and a larger byte-level BPE to optimize batch sequences and steps[12]. In previous studies, the BERT model was lacking to be used as an evaluation material and the results of hyperparameter tuning and dataset size have been proven. The results of the BERT modification to RoBERTa provide better performance [12]. The results of the BERT modifications are training older models with larger batches and having more data, deleting the next sentence for prediction purposes, and dynamically changing the masking pattern applied to the training data. RoBERTa will give more optimal results than BERT because of these modifications. This has been proven by research using the RoBERTa architecture [12].

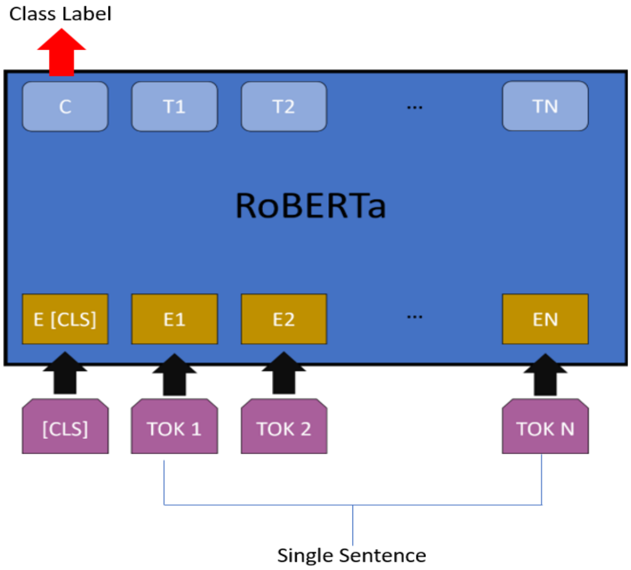

Fig. 2. RoBERTa Architecture

Fig 2 is the architecture of the RobertTa model. It can be seen in Fig 2 how RoBERTa works, namely, the RoBERTa model will receive input in a sentence where the sentence will be transformed into tokens so that it can become a valid input into the model. RoBERTa has 3 valid inputs, namely input_ids, attention_mask, and token_type_ids[16]. Based on the research that has been carried out, a machine will be developed that will detect hate speech in the comments column on Indonesian artists' Instagram. This machine will use the RoBERTa method in classifying. And will use the Confusion Matrix to determine the performance of the RoBERTa method in conducting this research.

After doing data splitting, there will be two datasets, namely training and testing. After the formation of the two datasets, encoding will be carried out for each dataset. In the RoBERTa model, a pre-trained model is required to perform an encoding on the dataset. In this research, the RoBERTa pre-trained model of cahya will be used. The pre-trained model uses masked language modeling. This model was previously trained using Indonesian Wikipedia and this model will detect the level of accuracy of a sentence, paragraph, or word. In the encoding process, there will be three inputs that will be processed in the model, namely input_ids, attention_mask, and token_type_ids. At this stage, comments will be tokenized in the RoBERTa modeling process. This process will be adapted to the RoBERTa pre-trained model in which the model attempts to detect a sentence from the comment. If the model understands the words from the comments, the comments will become one unit and if the model does not understand the words in the comments, the words will be separated.

Table 6 shows that in the RoBERTa model, there will be a tokenize where it changes a sentence to be separated in other words to be per word. This is a form of input_ids. Apart from the input_ids form, there will be a form of attention_mask which functions if in the sentence there is a missing word or the length of the word is not the same then an attention_mask is needed. Apart from these two inputs, there will be an input token_type_ids where this input will select whether these words will be made into a sentence or not.

TABLE 6
TOKENIZE ROBERTA

Stages	Result
Real Data	semua komentar isinya hujatan seru bacanya komentar nya seperti baca novel saja begitu kan
Tokenize RoBERTa	['semua','Ġkomentar','Ġisinya','Ġh','uj','atan','Ġseru','Ġba','canya','Ġkomentar','Ġnya','Ġseperti','Ġbaca','Ġnovel','Ġsaja','Ġbegitu','Ġkan']

TABLE 7
AFTER TOKENIZE ROBERTA

Stages	Result
Tokenize RoBERTa	['semua','Ġkomentar','Ġisinya','Ġh','uj','atan','Ġseru','Ġba','canya','Ġkomentar','Ġnya','Ġseperti','Ġbaca','Ġnovel','Ġsaja','Ġbegitu','Ġkan']
Input RoBERTa	{'input_ids': array([[0, 283, 3391, ..., 1, 1, 1], [0, 21779, 272, ..., 1, 1, 1], [0, 31738, 269, ..., 1, 1, 1], ..., [0, 2031, 400, ..., 1, 1, 1], [0, 81, 4248, ..., 1, 1, 1], [0, 7018, 283, ..., 1, 1, 1]], dtype=int32), 'attention_mask': array([[1, 1, 1, ..., 0, 0, 0], [1, 1, 1, ..., 0, 0, 0], [1, 1, 1, ..., 0, 0, 0], ..., [1, 1, 1, ..., 0, 0, 0], [1, 1, 1, ..., 0, 0, 0], [1, 1, 1, ..., 0, 0, 0]], dtype=int32), 'token_type_ids': array([[0, 0, 0, ..., 0, 0, 0], [0, 0, 0, ..., 0, 0, 0], [0, 0, 0, ..., 0, 0, 0], ..., [0, 0, 0, ..., 0, 0, 0], [0, 0, 0, ..., 0, 0, 0], [0, 0, 0, ..., 0, 0, 0]], dtype=int32)}

Table 7 is an example of a valid input for the RoBERTa model. There are three inputs, namely input_ids, attention_mask, and token_type_ids which will be processed in the RoBERTa model. After the input from the RoBERTa model has been formed, a RoBERTa model will be built using pretrained. The RoBERTa model will fine tune the pretrained so that it can classify. The wrapped input_ids, attention_mask, and token_type_ids will be processed in the base model which will output the last_hidden_state layer as output. This layer is where the embeddings of the ten layers are stored. In the RoBERTa model process, there will be an additional output layer, among which there is a dense layer that acts as a classifier and receives input from the neurons of the previous layer. Next, there is a layer that is used to avoid overfitting the neural network, that layer is the dropout layer. There are over 225,000,000 parameters in the RoBERTa model. After the RoBERTa model is formed, then it enters the Indonesian commentary classification stage which has been labeled positive comments and negative comments.

D. Evaluation

In 2018 a study was conducted regarding hate speech. In this study they used a confusion matrix to measure the performance of the model they used [15]. In this study, to measure the performance of a model, we use a confusion matrix. In the confusion matrix there are accuracy, precision, recall and F1 to measure the performance of the model [16]. The confusion matrix has four basic characteristics to measure accuracy, precision, recall and F1.

1) True Positive (TP) is the number of predictions that are correct when the predictions are correct.

2) True Negative (TN) is the number of predictions that are wrong when the predictions are correct.
3) False Positive (FP) is the number of predictions that were correct when the predictions were wrong.
4) False Negative (FN) is the number of predictions that are wrong when the predictions are wrong[17].

This research uses a confusion matrix in defining or measuring the performance of RoBERTa in the form of a table. The table is the result of RoBERTa's performance in classification, there will be four measurements in the table, namely true positive, false negative, true negative, and false negative. With these four measurements, we will get the accuracy, precision, recall, and F1-score values. To get these values, formulas will be used for each assessment parameter. Here are the formulas that will be used:

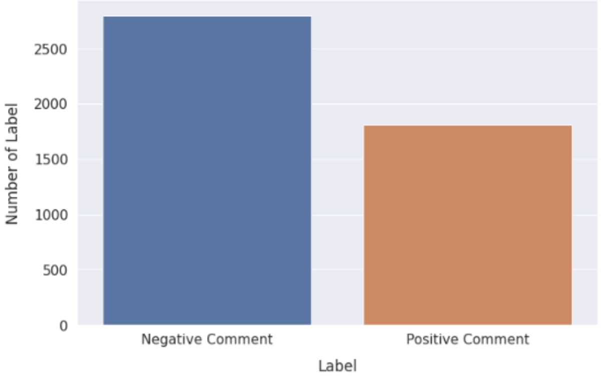

Fig. 3. Design Histogram Labelling

$$Accuracy = \frac{TP + TN}{TP + TP + FP + TN + FN} \tag{1}$$

$$Precision = \frac{TN}{FN + TN} \tag{2}$$

$$Recall = \frac{TN}{FP + TN} \tag{3}$$

$$F1 = \frac{2.(Recall.Precision)}{Recall + Precision} \tag{4}$$

III. Result

In the testing process, the dataset used contained usernames, comments, and labels obtained a total of 4602 comments had been collected. The contents of these comments have been labeled positive and negative. There are 2793 comments with negative labels and 1809 comments with positive labels.

In this study, the researcher will compare the results of the accuracy of full PreProcessing (Cleaning, Case Folding, Normalization, Tokenization, and Stemming) with PreProcessing using only (Cleaning, Case Folding, and Normalization) or in other researchers' words compare sentences that are said to be omitted a lot and not completely eliminated. This of course affects the battery yield of the Roberta model.

A. Full-PreProcessing Scenario Test Results

In the first scenario using Cleansing, Case Folding, Normalization, Tokenization, and Stemming to manage the collected datasets. In this scenario apart from PreProcessing the other things that will be tested are test_size which is changed from 0.1 to 0.3 and batch_size which is changed from 8, 16, and 32. To see how accurate the results of the accuracy of the RoBERTa model are, there will be 10 epochs to compare the accuracy results. Following are the results of the average accuracies for each different parameter:

TABLE 8
RESULT FULL-PREPROCESSING

Test_Size	Batch_size	Data Train	Average Accuracy
0,1	8	512	84,21%
0,1	16	256	83,79%
0,1	32	128	83,05%
0,2	8	455	84,81%
0,2	16	228	84,18%
0,2	32	114	81,99%
0,3	8	399	82,94%
0,3	16	200	82,08%
0,3	32	100	82,38%

Table 8 is the result of the analysis of the RoBERTa model which uses full preprocessing. Table 8 shows the average output of the accuracy that has been processed in the RoBERTa model and the accuracy results where each parameter is changed to get different accuracy values.

B. Non Full-PreProcessing Scenario Test Results

The second scenario is more or less the same as the first scenario but only uses Cleansing, Case Folding, and Normalization to manage the collected datasets. Because not using Full-PreProcessing will affect the accuracy of the RoBERTa model. Because RoBERTa will read all the conjunctions and stuff. In this scenario apart from PreProcessing the other things that will be tested are test_size which is changed from 0.1 to 0.3 and batch_size which is changed from 8, 16, and 32. To see how accurate the results of the accuracy of the RoBERTa model are, there will be 10 epochs to compare the accuracy results. The following are the results of the average accuracies for each different parameter:

TABLE 9
RESULT NON FULL-PREPROCESSING

Test_Size	Batch_size	Data Train	Average Accuracy
0,1	8	512	84,16%
0,1	16	256	85,09%
0,1	32	128	84,87%
0,2	8	455	84,16%
0,2	16	228	84,26%
0,2	32	114	83,02%
0,3	8	399	81,49%
0,3	16	200	82,30%
0,3	32	100	83,95%

Table 9 is the result of an analysis of the RoBERTa model which does not use full preprocessing. Table 9 shows the average output of the accuracy that has been processed in the RoBERTa model and the accuracy results where each parameter is changed to get different accuracy values.

C. Analysis of First Scenario Test Results

Analysis related to the first scenario of doing modeling using full-PreProcessing is that it can be seen that the highest average accuracy it has is 84,81%, which means that RoBERTa is able to make predictions with data conditions that have been tokenized twice because, in the full-preprocessing stage, it uses tokenization and the RoBERTa modeling also uses tokenization, therefore RoBERTa can detect sentences even though the tokenization process twice. RoBERTa is also still able to make predictions on comments or sentences where the conjunctions are removed during pre-processing. It can also be seen that with different batch_size numbers, the accuracy numbers have slight differences, but the differences are not too great. Changing the test_size and batch_size, just shows the difference in accuracy values where if the tezt_size gets bigger and the batch_size gets bigger, the resulting data or val_accuracy value can be even lower. For example, test_size = 0.2 and batch_size = 32 have 114 data train results, and the lowest average accuracy is 81,99%. Things that affect the results of this accuracy are the small amount of data trained and the large number of batch_sizes so that the val_accuracy can reach 81.99%. This can be seen in the results of the accuracy of each batch_size which is changed, it will affect the results of the accuracy. For batch_size=8 it has the highest average accuracy of 84,81% while batch_size=16 has the highest average accuracy value of 84,18% and batch_size=32 has the highest average accuracy value of 83.05%. This shows that the smaller the number of train data, the higher the accuracy will be obtained. Because the smaller the batch_size, the faster the convergent algorithm will be, but the greater the computational noise will be. Apart from the batch_size that affects the results of the accuracy is test_size. Test_size also affects the results of accuracy because the train data used is reduced, the higher the test_size the data will be less. In this study, the researchers changed the test_size to 0.1, 0.2, and 0.3. For test_size 0.1 and batch_size 8 have data train 512, for test_size 0.1 and batch_size 16 have data train 256 and test_size 0.1 and batch_size 32 have data train 128. For test_size 0.2 and batch_size 8 have data train 455, for test_size 0.2 and batch_size 16 it has data train 228 and test_size 0.2 and batch_size 32 have data train 114. For test_size 0.3 and batch_size 8 it has a data train of 399, for test_size 0.2 and batch_size 16 it has a data train of 200 and test_size 0.2 and batch_size 32 has a data train of 100.

D. *Analysis of First Scenario Test Results*

The results of the analysis of the second scenario using pre-processing only up to the normalization stage show that the accuracy value is higher than using full pre-processing. The highest average accuracy value in the second scenario is 85.09%, this shows that RoBERTa will work more optimally if the data is not processed with full preprocessing, because the RoBERTa model detects all the words in the sentence, therefore conjunctions, conjunctions, and others are very important. important in RoBERTa modeling. In each epoch you can also see a random decrease in numbers, this is because the results of the val_accuracy generated from each epoch will be different, it could go up and it could go down. This is influenced by the inconsistent model in calculating val_accuracy. But the value of accuracy will continue to increase until it approaches number one. The output of the second scenario for batch_size=8 has the highest average accuracy of 84.16%, while batch_size=16 has the highest average accuracy value of 85.09% and batch_size=32 has the highest average accuracy value of 84.87%. Apart from batch_size which affects the results of test_size accuracy, it also affects the results of accuracy because less train data is used when changing test_size. In this second scenario, the researcher also changes the test_size to 0.1, 0.2, and 0.3. For test_size 0.1 and batch_size 8 have data train 512, for test_size 0.1 and batch_size 16 have data train 256 and for test_size 0.1 and batch_size 32 have data train 128. For test_size 0.2 and batch_size 8 have data train 455, test_size 0.2 and batch_size 16 have data train 228 and for test_size 0.2 and batch_size 32 have data train 114. For test_size 0.3 and batch_size 8 have a data train of 399, test_size 0.2 and batch_size 16 have a data train of 200 and test_size 0.2 and batch_size 32 have a data train of 100.

IV. Conclusion

This research detects hate speech comment that is labeled positive and negative. The dataset is obtained by determining 3 Indonesian artists who have a lot of blasphemy or hate speech in the comments column of their Instagram posts. And in this study, the dataset will be managed using Full-PreProcessing and non Full-PreProcessing to conduct a study comparing the two things. In addition to being able to see the comparison of the highest accuracy value between Full-PreProcessing and non Full-Preprocessing, to measure the results of the performance of the model that has been built, the confusion matrix can be used as a reference for measuring performance. From the test scenarios that have been carried out, it can be seen that Full-PreProcessing and non Full-PreProcessing greatly affect the accuracy value of the RoBERTa model. Because can be seen from the scenario that has been done, there is a difference in accuracy value where the value of Full-PreProcessing is lower than that of non-Full-PreProcessing. Because in the RoBERTa model, all words in the sentence or comment will be detected such as conjunctions and others. If one of the words is missing in a sentence this can reduce the accuracy results. Suggestions for further research are to do more varied training data so as to get varying accuracy values so that these values can be compared between one value and another. In addition, more exploration is related to parameters that affect the performance or accuracy results of the RoBERTa model.

REFERENCES

[1] D. Kusumasari and S. Arifianto, "Makna Teks Ujaran Kebencian Pada Media Sosial," *Jurnal Komunikasi*, vol. 12, no. 1, p. 1, Jan. 2020, doi: 10.24912/jk.v12i1.4045.

[2] "Digital around the world in April 2020 - We Are Social UK." https://wearesocial.com/uk/blog/2020/04/digital-around-the-world-in-april-2020/ (accessed Jan. 29, 2023).

[3] "BUDAYA BERKOMENTAR WARGANET DI MEDIA SOSIAL: UJARAN KEBENCIAN SEBAGAI SEBUAH TREN – Environmental Geography Student Association." https://egsa.geo.ugm.ac.id/2022/02/06/budaya-berkomentar-warganet-di-media-sosial-ujaran-kebencian-sebagai-sebuah-tren/ (accessed Jan. 29, 2023).

[4] C. Bagdon, "Profiling Spreaders of Hate Speech with N-grams and RoBERTa Notebook for PAN at CLEF 2021," 2021. [Online]. Available: http://ceur-ws.org

[5] B. P. I. B. & S. C. Putra, "Deteksi Ujaran Kebencian dengan Menggunakan Algoritma Convolutional Neural Network pada Gambar," 2018.

[6] Oryza Habibie Rahman, Gunawan Abdillah, and Agus Komarudin, "Klasifikasi Ujaran Kebencian pada Media Sosial Twitter Menggunakan Support Vector Machine," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 1, pp. 17–23, Feb. 2021, doi: 10.29207/resti.v5i1.2700.

[7] Y. Suryani, R. Istianingrum, and S. U. Hanik, "Linguistik Forensik Ujaran Kebencian terhadap Artis Aurel Hermansyah di Media Sosial Instagram," *BELAJAR BAHASA: Jurnal Ilmiah Program Studi Pendidikan Bahasa dan Sastra Indonesia*, vol. 6, no. 1, pp. 107–118, Mar. 2021, doi: 10.32528/bb.v6i1.4167.

[8] M. Kurnia Maulidina and E. Itje Sela, . "ANALISIS SENTIMEN KOMENTAR WARGANET TERHADAP POSTINGAN INSTAGRAM MENGGUNAKAN METODE NAÏVE BAYES CLASSIFIER DAN TF-IDF (Studi Kasus: Instagram Gubernur Jawa Barat Ridwan Kamil)."

[9] S. P. P. B. Batara, "Deteksi Ujaran Kebencian Dalam Bahasa Indonesia Pada Kolom Komentar Instagram Dengan Metode Klasifikasi Deep Neural Network," 2019.

[10] A. Briliani, "Deteksi Ujaran Kebencian Dalam Bahasa Indonesia Pada Kolom Komentar Instagram Dengan Metode Klasifikasi K-Nearest Neighbor," 2019.

[11] E. Erizal, "Deteksi Ujaran Kebencian Dalam Bahasa Indonesia Pada Kolom Komentar Instagram Dengan Metode Klasifikasi Maximum Entropy," 2019.

[12] Y. Liu *et al.*, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," Jul. 2019, [Online]. Available: http://arxiv.org/abs/1907.11692

[13] "Glossary." https://huggingface.co/docs/transformers/glossary (accessed Jan. 29, 2023).

[14] "Dasar Text Preprocessing dengan Python | by Kuncahyo Setyo Nugroho | Medium." https://ksnugroho.medium.com/dasar-text-preprocessing-dengan-python-a4fa52608ffe (accessed Jan. 29, 2023).

[15] S. Zimmerman, C. Fox, and U. Kruschwitz, "Improving Hate Speech Detection with Deep Learning Ensembles." [Online]. Available: https://www.economist.com/news/europe/21734410-

[16] "Mengenal Accuracy, Precision, Recall dan Specificity serta yang diprioritaskan dalam Machine Learning | by Resika Arthana | Medium." https://rey1024.medium.com/mengenal-accuracy-precission-recall-dan-specificity-serta-yang-diprioritaskan-b79ff4d77de8 (accessed Jan. 29, 2023).

[17] "Understanding Confusion Matrix | by Sarang Narkhede | Towards Data Science." https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62 (accessed Jan. 29, 2023).