Internasional Journal of Integrative Sciences (IJIS)
Vol.2, No.7, 2023: 1015-1030

Sentiment Analysis of Negative Comments on Social Media Using Long Short-Term Memory (LSTM) Method with TensorFlow Framework

Iwan Giri Waluyo[1], Juwono[2*]
Pamulang University
Corresponding Author: Juwono juwonoindonesia@gmail.com

A R T I C L E I N F O

Keywords: Long Short Term Memory, Sentiment Analysis, Social Media, Natural Language Processing

Received : 20 May
Revised : 22 June
Accepted: 19 July

©2023 Waluyo, Juwono: This is an open-access article distributed under the terms of the Creative Commons Atribusi 4.0 Internasional.

A B S T R A C T

There are still many unidentified negative comments on social media that have a negative impact on others' mental and physical well-being. Therefore, sentiment analysis is needed to filter and identify such types of comments, especially on social media. This research aims to analyze and classify unidentified negative comments spread across social media. Sentiment analysis and comment classification are performed using 7773 comments in the Indonesian language. The comments are then visualized using an embedding projector, which gives satisfactory results in classifying words in the comments, where words with negative or positive sentiments are clustered closely together. The model employed in this study is the Long Short Term Memory (LSTM) model, which achieved an accuracy rate of 77.70% and a validation accuracy of 85.20%. The trained model is then used for testing purposes, employing directly collected comments from social media, which give satisfactory results

DOI: https://doi.org/10.55927/ijis.v2i7.4990
ISSN-E: 2985-9050
https://journal.formosapublisher.org/index.php/ijis

INTRODUCTION

The rapid development of the internet has made many social media sites appear and develop, various types of social media applications available on the internet are used by various people in the world to interact with each other, be it by sending messages or commenting on social media. Information and communication which is a primary need for humans in this era of technological development makes social media grow very rapidly. According to We are social (Hootsuite, 2022) , the development of social media usage at the end of January 2022 reached 4.62 billion users, an increase of about 10.1 percent compared to the previous year. The increasing number of social media users on the internet means that the amount of existing content will increase.

In social media, users who create content tend to want to be responded to or get recognition from other users either in the form of responses such as like to the content, or in the form of comments on the content. Sometimes comments and open discussions that exist on social media can trigger debates caused by some irresponsible people with negative comments.

Negative comments can also cause various other problems on social media such as cyberbullying, sexual harassment, or other negative impacts that can be felt directly mentally or physically, especially in adolescence. As a result of these actions, some people will stop giving opinions or try to avoid debates on social media that lead to unhealthy and unfair discussions. It becomes very difficult for social media apps and online communities to facilitate fair conversations and people feel restricted from commenting.

The problem of negative comments in the comment field is very important to be studied in text processing techniques, from several comments scattered in social media applications, sentiment analysis is needed to filter and identify the type of comment. Sentiment analysis on comments is done to find out comments that are negative and positive.

Through sentiment analysis using deep learning, it is expected to help find out and classify negative comments that have not been identified and spread on social media, reduce the negative impact caused by negative comments on social media, and detect negative comments automatically so that the discussion space on social media becomes more effective and comfortable for everyone.

LITERATURE REVIEW

Related Work

Text processing is the most important thing for managing text in order to provide useful information. Research by Sharma and Patel (Sharma & Patel, 2018) conducted classification using several methods to find the best solution in text processing, these methods include Neural Network (NN), Convolutional Neural Network (CNN) for Text Classification, and LSTM. The analysis found that LSTM performs better than CNN and NN in terms of accuracy and time performance given in each epoch. For accuracy, LSTM has more than 98% accuracy at the 7th epoch onwards, while CNN and NN have about 97% accuracy at the 7th epoch onwards. This is what makes LSTM more favourable for use in text processing and classification. Classification of toxic comments has also been done a lot, such as research conducted by Zaheri et al (Zaheri, Leath, & Stroud,

2020). Where they conducted toxic classification using LSTM method. The model can classify several negative comments such as insults, verbal sexual behaviour and defamation whose dataset comes from Wikipedia Talk pages. The classification results show good accuracy with more than 70%. Meanwhile, research on sentiment analysis has been conducted by Cahyadi et al (Cahyadi, Damayanti, & Aryadani, 2020) on comments on Instagram social media using the Recurrent Neural Network (RNN) method with Long Short Term Memory (LSTM). The result of this research is a system that can classify sentiment with a testing accuracy rate of 65% and a training accuracy rate of 79.64%. Research on sentiment analysis using LSTM algorithm has been conducted by Mudding et al (Mudding & Karim, 2022) using Indonesian comment-type data from Twitter social media.

TensorFlow Data Pipelines

In this research, the model is trained using the TensorFlow model, so it is necessary to prepare TensorFlow Data Pipelines so that the processing process on the dataset becomes structured, efficient and easy to repeat. The process involves several stages, namely map or tensor slice, cache, shuffle, batch, and prefetch as shown in Figure 1.

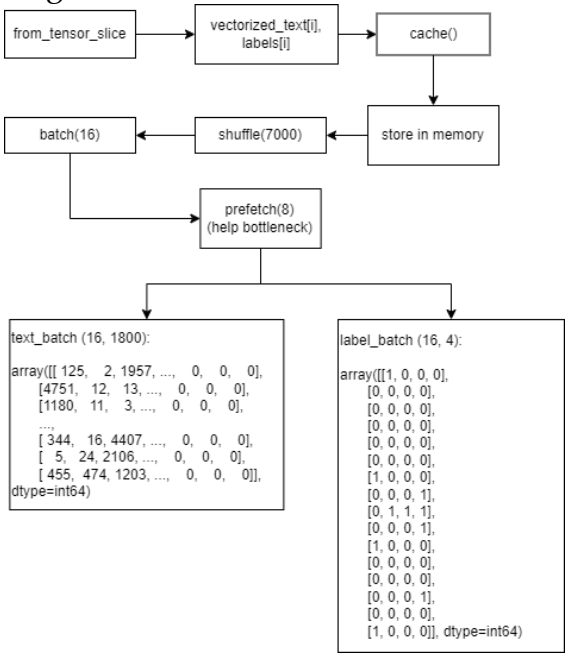

Figure 1. TensorFlow Data Pipelines

The from_tensor_slices() method of tf.data.Dataset is used to create a dataset of vectorised_text and labels tensors. Vectorised_text is a tensor that contains text that has been tokenized or has gone through a tokenization process while labels is a tensor that contains labels that correspond to the text. This method creates a dataset by slicing the input tensor in the first dimension so that each pair of text and its label (vectorised_text[i], labels[i]) becomes a dataset element. Furthermore, the cache() method is used to cache the dataset after the first run, this allows the data to remain in memory after the first read, thus avoiding repeated reads from the data source during model training or

evaluation. This is useful if the data used can be fully loaded into memory. After the dataset is stored into the cache to avoid repetition of the next reading the dataset goes through a shuffle process. The batch(16) method is used to group the dataset elements into batches of size 16. Each batch will consist of 16 pairs (vectorised_text[i], labels[i]) thus allowing parallel training on the GPU (Graphics Processing Unit) and increasing processing efficiency. Furthermore, the prefetch() method is used to batch the data into memory before it is needed during training.

Network Architecture Models

In machine learning, a model refers to a mathematical or statistical representation of a phenomenon or system being studied (Aggarwal, 2018). The TensorFlow model consists of a series of interconnected layers and each layer contains a set of processing units called neurons or nodes (Scarpino, 2018). The model is created using several layers or layers with a structure using TensorFlow, namely a Sequential neural network. The Sequential model is a model that contains a series of layers that are connected sequentially. Some of the layers used are the Word Embedding layer, Bidirectional LSTM layer, and Dense layer as shown in Figure 2.

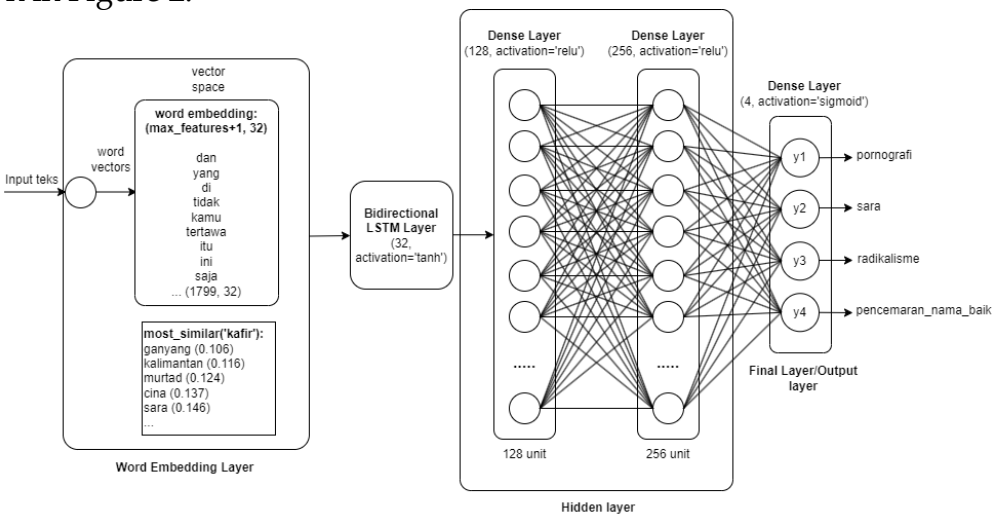

Figure 2. Network Architecture Models

The Word Embedding layer is the first layer and is used to convert the word representation in the text into a vector representation with lower dimensions. The max-features+1 parameter specifies the vocabulary size or the number of unique words used in the data, and the number 32 specifies the dimension of the embedding vector. To see the results of the word representation, in this study, the TensorFlow Embedding Projector is used to visualise the word representation in vector form in multi-dimensional space and see the context of words that are close to each other as shown in Figure 3.

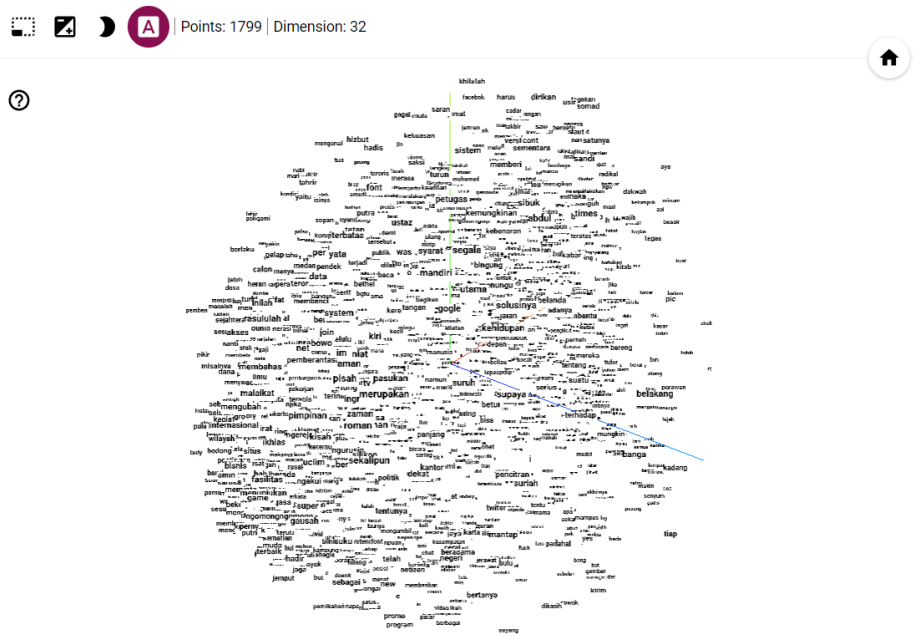

Figure 3. Word Embedding Projector

The LSTM layer is used to learn the patterns in the sequence data, in this case, the text that appears from the comments in the dataset. With the Bidirectional layer, the model can process forward and backward data sequences simultaneously, thus gaining more contextual information. Dense layer as a Fully Connected Layer receives input from each neuron in the previous layer and produces an output that is multiplied by its weight and added with a bias. A ReLU (Rectified Linear Unit) activation function is used which will convert any negative value to zero and maintain positive values unchanged, if the input is a positive value, then the output of the ReLU function will be equal to the input. However, if the input is a negative value, then the output of the ReLU function will be zero (Zhang, Lipton, Li, & Smola, 2019).

Model Evaluation

Model evaluation is the process of assessing and measuring the performance of a machine learning or deep learning model. The purpose of model evaluation is to understand the extent to which the model is successful in learning and generalizing patterns from the given data (Putra, 2020). In this research, the model will be evaluated using precision, recall, and accuracy metrics. The parameters used are TP (True Positive), FN (False Negative), TN (True Negative), and FP (False Positive) (Singh & Manure, 2020). tensorflow.keras.metrics module is used to make model evaluation by using Precision, Recall, and CategoricalAccuracy (accuracy) metrics provided by the module. Formulas for calculating precision, recall, and accuracy can be seen in Table 1.

Table 1. Model Evaluation Formula

Metrik	Rumus
Precision	$\frac{TP}{TP + FP}$
Recall	$\frac{TP}{TP + FN}$
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$

METHODOLOGY

The concept of methodology in this study are shown in Figure 4. Based on the illustrations in Figure 4, the core steps of this research are data construction which includes the stages of data collection from social media, data analysis, and preprocessing.

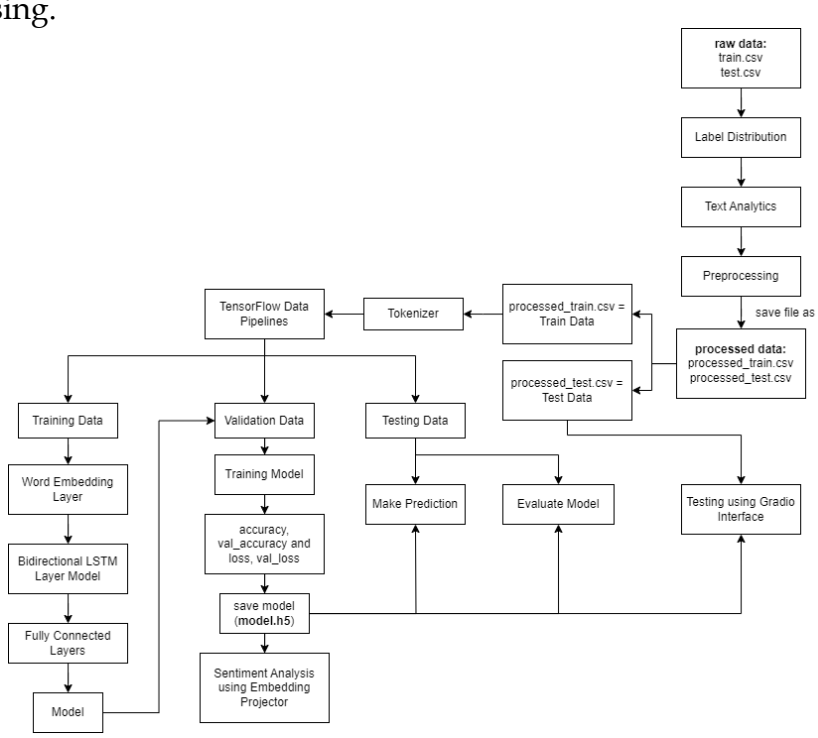

Figure 4. Concept of Methodology

Data Source

The data used in this study comes from several comments on social media such as Twitter, Instagram, and Kaskus collected by the Indonesia Social Media Text Toxicity Dataset from 2019 to 2021. The Indonesia Social Media Text Toxicity Dataset is a dataset containing around 7,773 example sentences inspired by the Toxic Comment Classification Challenge on Kaggle which is an online data science community platform, something similar is done using data from social

media in Indonesia where the sentences come from several user posts on social media and then categorized into positive sentences and negative sentences in Indonesian. Negative sentences consist of 4 (four) types, namely pornography, SARA (ethnicity, race, religion, intergroup), radicalism, and defamation, while a positive sentence is a sentence that does not contain negative meaning. The dataset is then called raw data or raw data that has not been preprocessed with a total of 7,773 sentences. The raw data dataset is also divided into two datasets, namely train data and test data. Train data is 90% of the total raw data or 6,995 comments while test data is 10% of the total raw data or 778 comments.

Data Analysis

After knowing the number of sentences in the dataset, the data is then analyzed by giving labels to each comment. After all the data has a label that indicates the category of the sentence, the next step is to perform label distribution (Kulkarni & Shivananda, 2019). The results of the label distribution can be seen in Figure 5 that the dataset has a fairly well-distributed number of each negative sentence category or label and the number of sentences labeled 0 (zero) or positive sentiment is more than those labeled 1 (one) or negative sentiment. The highest positive labels are owned by sentences categorized as SARA and the highest negative labels are owned by sentences categorized as defamation.

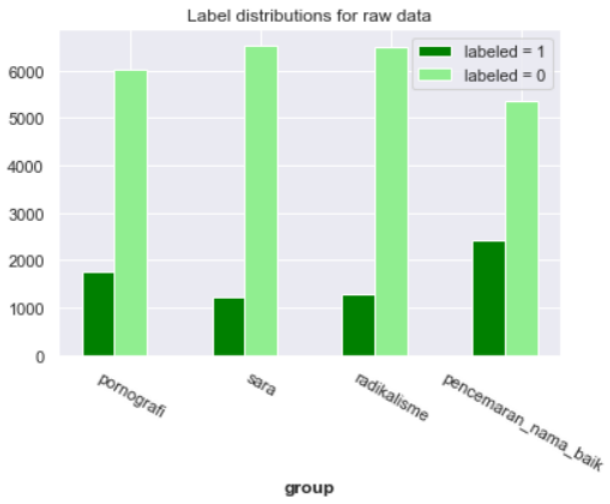

Figure 5. Label Distribution Graph for Sentences in the Raw Data Dataset

After the data is collected in the form of raw data, the data is then divided into train data as much as 90% and test data as much as 10%. Train data aims to train data that will build a model formed with a size of 70% training data, then the model is validated to avoid overfitting by using 20% of the training data and the remaining 10% becomes testing data to test the performance of the model that has been trained. Then the data analysis process is carried out using train data as the main dataset through visualization of the correlation matrix. As seen in the correlation matrix in Figure 6, there are several comments that are correlated with each other, for example, there is a possibility that comments categorized as SARA are also radicalism, or there is a possibility that comments categorized as

defamation are also SARA or vice versa so that the dataset is included in the machine learning task called Multi-label Classification.

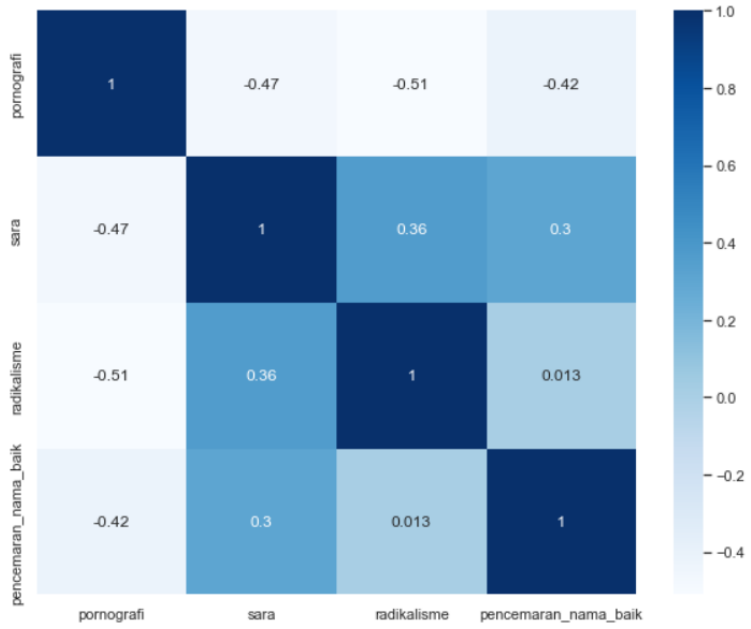

Figure 6. Correlation Matrix between Labels

Preprocessing

The following are some of the preprocessing steps used to process text in model training:

1. Case folding, is the process of converting all letters in a text or sentence into lowercase or uppercase letters.
2. Translate text based on emoticons. Since the sentences in the dataset must be able to recognize various types of emoticons, a text file (.txt) format dataset named emoticons.txt was collected which contains various types of emoticon formats and their meanings.
3. Remove excessive white space and new lines. Some sentences in the dataset contain excessive white space and new lines, which automatically affects the composition and meaning of the sentence.
4. Remove symbols, numbers, punctuation, and URL formatting. A sentence usually contains symbols, numbers, or punctuation marks. All of these need to be removed so that the sentences used in the training process can focus on the important words only and can reduce the feature dimension because a large number of features can affect model performance and computation time and the presence of symbols, numbers or punctuation often does not contribute significantly to the meaning of the text (Lane, Howard, & Hapke, 2019).
5. Removing repetitive characters. Some sentences found in the dataset also contain some repetitive characters that need to be removed to avoid sentence meaning errors and make the data analysis process more efficient.

6. Change slang words. The slang words must be converted into standard sentences in order to obtain a consistent representation in the analysis using the lemmatization.
7. Removing social media formatting. Social media formatting must be removed as it will impact the performance of the model, the process of removing social media formatting on sentences in the dataset uses regular expression syntax.
8. Tokenization, is the process of converting text or sequences of characters into smaller discrete units called tokens. Tokenizers aim to create a more structured structural representation of text so that it can be used in various language processing tasks (Lane, Howard, & Hapke, 2019).

RESULTS

Results of Training Models

Based on the model training process carried out in as many as 15 epochs, the lowest loss value on accuracy is 0.0305 at the 12th epoch, the lowest loss value on accuracy validation (val_accuracy) is 0.0220 at the 12th epoch, the highest accuracy value is 77.70% at the 12th epoch and the highest accuracy validation value (val_accuracy) is 85.20% at the 12th epoch. Based on the results of the values obtained, it can be seen that the model is good enough and does not experience overfitting. Comparison of training and validation accuracy values and training and validation loss values can be seen in Figure 7.

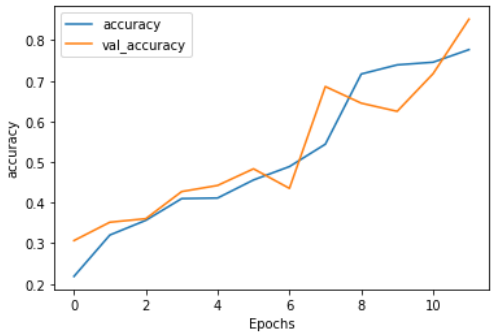

Figure 7. Training and Validation Accuracy Values

Evaluation of Model Test Data

As shown in Table 3, the classification evaluation in this study yielded good results based on the reliable accuracy value of 85.38%. The precision value of 98.72% is considered quite good for predicting sentiment, and the recall value of 97.17% indicates a satisfactory level of sensitivity for the model's performance. The performance of the model tested on data testing produces an accuracy that is not too much different from the results of the model made, this happens because the data used in data testing may not be so complicated, and with the right portion of the division in the data pipelines process.

Table 2. Model Evaluation Results

Evaluation	Result
Precision	98.72%
Recall	97.17%
Accuracy	85.38%

Sentiment Analysis through Word Visualization

Sentiment analysis is carried out using the embedding projector as shown in Figure 8. An example of the word or label "*kebaikan*" is used which shows that there are words that are interconnected and semantic with the word "*kebaikan*". The greater the Nearest Point value, the closer and more positive sentiment relationship with the word. In this positive class visualisation can be used as a reference regarding what kind of words or comments contain positive sentiment properties or do not mean negative.

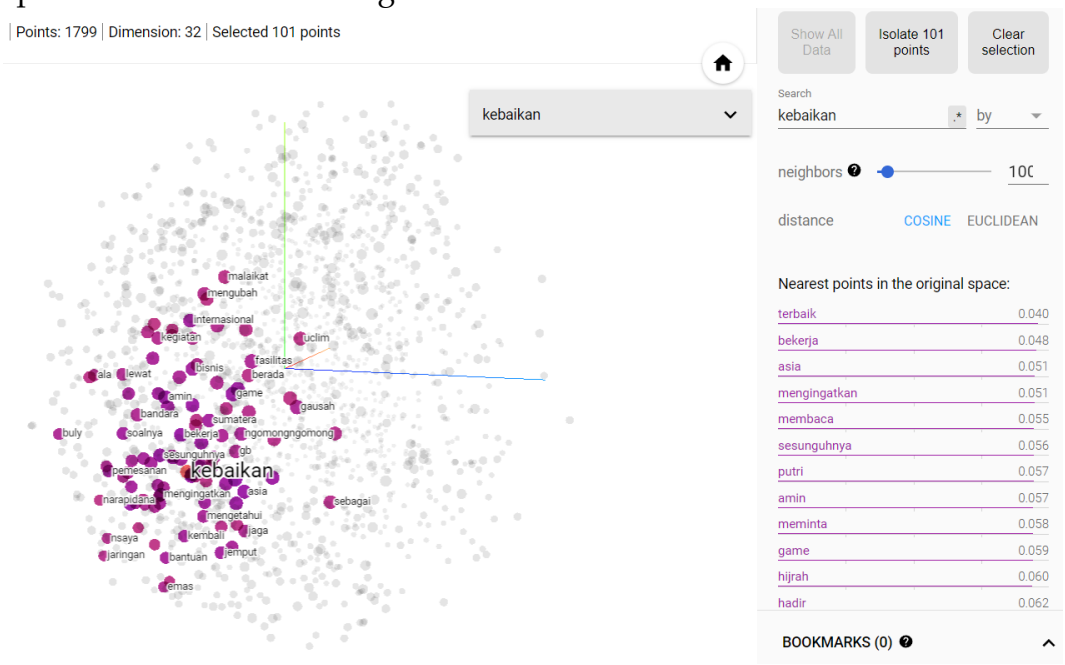

Figure 8. Embedding Projector in Positive Class

Meanwhile, in Figure 9, the visualization results for the negative class in the embedding projector are displayed. The example word or label used is "*kafir*," which appears to be quite dominant and shows similarities with several related or semantically connected words. From the visualization results of the negative class, patterns or characteristics regarding the types of words or comments that contain negative sentiment can be identified.

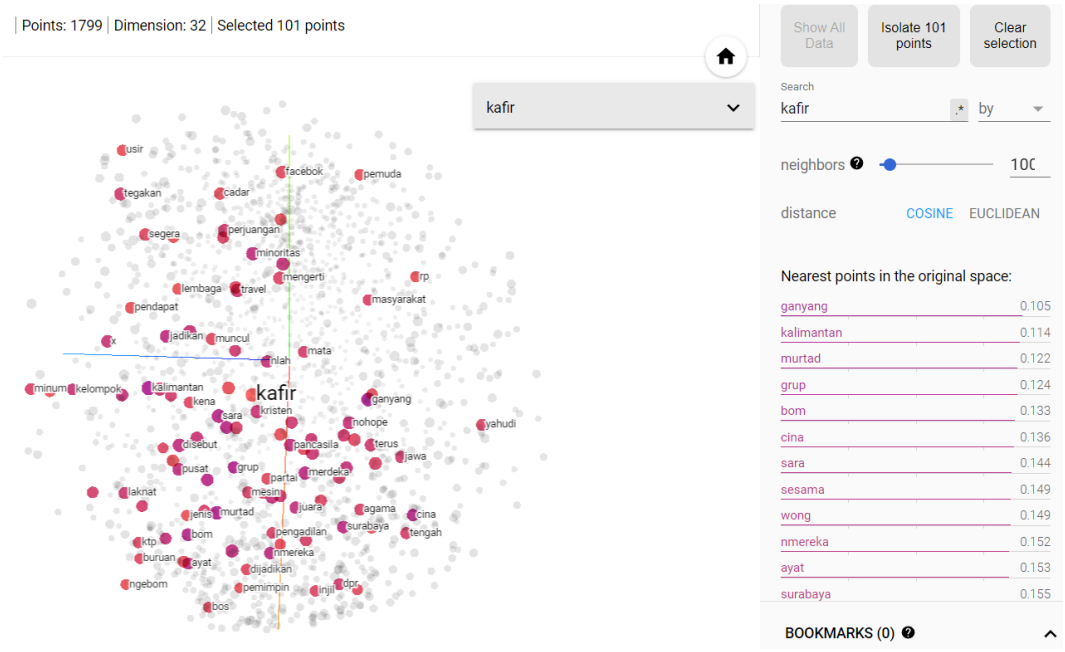

Figure 9. Embedding Projector in Negative Class

Make Predictions Based on Testing Data

The prediction is performed using testing data by providing a new text or word that has never been seen (unseen data) by the model. For example, the negative sentiment sentence "*Dasar Kafir gak tau malu!*" is given. This sentence is predicted using the model by creating input and label pairs from the testing data. Prior to this, the input of the sentence is converted into a numpy array and reshaped to match the format accepted by the model, which includes 4 classes: pornography, hate speech, radicalism, and defamation. The prediction results for the sentence indicate the following values: 0.00313 for the pornography class, 0.73855 for the hate speech (SARA) class, 0.00161 for the radicalism class, and 0.29828 for the defamation class. Based on these values, the sentence "*Dasar Kafir gak tau malu!*" can be classified as a negative sentence falling under the hate speech (SARA) category. From the prediction results, it can be observed that the trained model performs quite well in detecting and categorizing the sentence as a negative sentiment sentence.

DISCUSSION

Implementation Models Using Gradio

Gradio is a Python library used to build interactive user interfaces for machine learning models. Gradio provides components such as text input and buttons that can be used to test and demonstrate models using test data, specifically in this case, comments in the form of sentences. The trained model, saved as "model.h5," is directly tested using unseen data from the test data file ("processed_test.csv"), which includes various comments extracted from a social media post.

Using Test Data

The first test involved providing example sentences with positive or non-meaningfully negative classes taken from the test data through the Gradio interface. The generated output on the interface matched the labels in the test data, where all the negative class categories became "False," indicating that the sentences did not belong to any of the four negative classes (pornography, hate speech (SARA), radicalism, defamation), as seen in Figure 10.

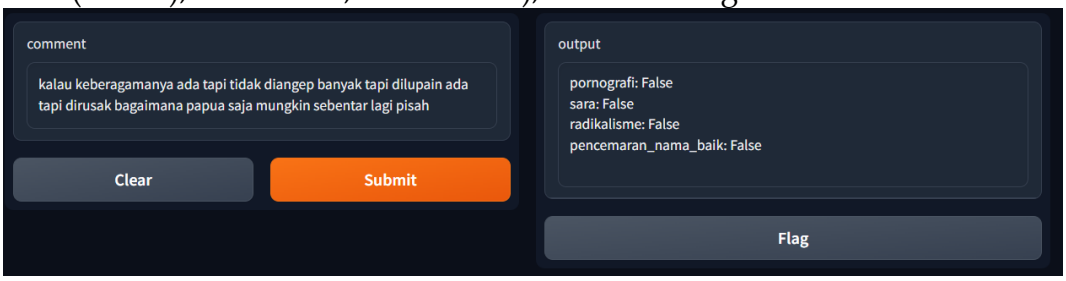

Figure 10. Testing on Sentences With a Positive Class.

The next test involved providing example sentences with negative classes taken from the test data. Based on the input sentences into the Gradio interface, the generated output matched the labels in the test data, specifically indicating negative sentences falling under the category of radicalism. In the testing process, the radicalism class was assigned a value of "True," as shown in Figure 11.

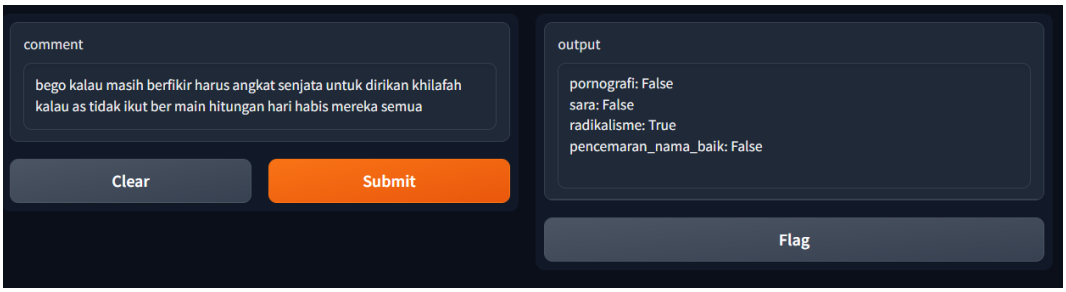

Figure 11. Testing on Sentences with a Negative Class

In the next test, example sentences with multi-label classes were used, meaning that the sentences could belong to more than one negative class. Based on the input sentences into the Gradio interface, the generated output did not exactly match the classes in the test data. The Gradio output indicated that the sentence belonged to the defasmation class only, labeled as "True." However, in the test data, the sentence had multiple negative class labels, specifically radicalism and defamation, as shown in Figure 12.

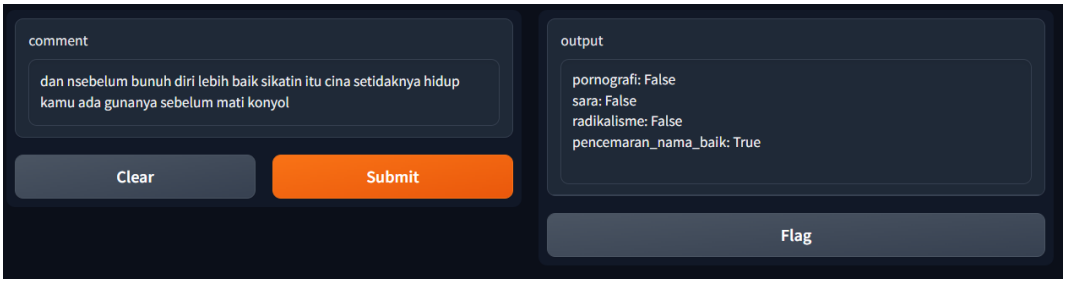

Figure 12. Testing on Sentences with Multi-label Negative Class

Using Posts on Social Media

Another test was conducted using a social media post directly taken from Twitter to assess the model's accuracy in classifying comments. The example post contained several comments, which underwent sentiment analysis and were tested using the model through the Gradio interface. The comment classification results yielded satisfactory performance, with negative class labels, specifically defamation, as shown in Table 3.

Table 3. Classification of Comments Based on Posts Using Gradio

No	Comment	*Output*
1	gaush caper ya anjing	pornografi: *False* sara: *False* radikalisme: *False* pencemaran_nama_baik: *True*
2	Apasih njenk, itu umur udh pas bego. Mksd lu, lu gapyear gitu? Ya gpp njir, byk kok yg gapyear, malah ada yg gapyear 2 tahunan. Angkatan gue kmren wkt 2021 byk yg msh umur 18-19 thn, dia udh umur 20 an and it's okay? Yg penting lu kuliah nder aplgi fk, byk loh yg ngincer	pornografi: *False* sara: *False* radikalisme: *False* pencemaran_nama_baik: *True*

CONCLUSIONS AND RECOMMENDATIONS

Conclusion

Based on the discussion on sentiment analysis of negative comments on social media using the Long Short-Term Memory method with the TensorFlow framework, along with the descriptions provided in the previous chapters, the following conclusions can be drawn:

1. Sentiment analysis and classification of negative comments were conducted using the embedding projector, yielding satisfactory results in classifying words within comment sentences, where words with negative or positive classes are closely related to each other.
2. The trained LSTM model performed quite well in predicting and classifying comments based on negative classes.
3. The training results of the model showed an accuracy rate of 77.70% and a validation accuracy of 85.20% at the 12th epoch. The trained model underwent evaluation with precision of 98.72%, recall of 97.17%, and accuracy of 85.38%.

Suggestion

The suggestions that can be put forward are as follows:

1. For further development, it is necessary to increase the dataset size so that the model can recognize a diverse range of semantic words in Indonesian comments. Additionally, adding comment data from other languages such as English or commonly used languages on social media is also important.
2. Building different models or network architectures during the training process can enhance the accuracy of the model.
3. For further development, the model results can be implemented into application programs such as an Application Programming Interface (API) or directly integrated into mobile or web-based social media applications to prevent, classify, and detect negative comments.

FURTHER STUDY

I hope that there will be further developments on this topic, because as a writer I know that the research I have conducted still has many shortcomings and further development is needed. The author hopes that future researchers will develop a better system than what I have done. Greetings from the author.

ACKNOWLEDGMENT

In writing this paper, the author experienced many obstacles, thanks to the guidance from Allah SWT and the help and guidance from various parties, all of these obstacles or difficulties could be resolved properly.

Furthermore, the author's thanks go to:

1. Dr. Pranoto, S.E., M.M. as chairman of the Pamulang University Sasmita Jaya Foundation.
2. Dr. E. Nurzaman AM, M.M., M.Si. as the Rector of Pamulang University.
3. Dr. H. Sarwani, M.T., M.M. as the Dean of the Faculty of Computer Science, Pamulang University.
4. Achmad Udin Zailani, S.Kom., M.Kom. as the Head of the Informatics Engineering Study Program, Pamulang University.
5. Iwan Giri Waluyo, S.Kom., M.Kom. as my Academic Advisor.
6. To my Parents who have supported and motivated me throughout my life.
7. All the Lecturers who have shared their knowledge during my studies at Pamulang University.
8. Relatives and Friends in the TPLE007 class who have provided moral support.
9. Fellow classmates, especially the 2018 batch of Informatics Engineering students, who have motivated me throughout my journey.

REFERENCES

Aggarwal, C. C. (2018). *Neural Networks and Deep Learning*. Switzerland: Springer International Publishing AG

Cahyadi, R., Damayanti, A., & Aryadani, D. (2020, Februari). Recurrent Neural Network (RNN) Dengan Long Short Term Memory (LSTM) Untuk Analisis Sentimen Data Instagram. *Jurnal Informatika dan Komputer (JIKO), V*(1), 1-9.

Hootsuite. (2022, January 26). *Digital 2022: Another Year of Bumper Growth*. Retrieved from We are social: https://wearesocial.com/uk/blog/2022/01/digital-2022-another-year-of-bumper-growth-2/

Kulkarni, A., & Shivananda, A. (2019). *Natural Language Processing Recipes: Unlocking Text Data with Machine Learning and Deep Learning using Python.* India: Apress.

Lane, H., Howard, C., & Hapke, H. M. (2019). *Natural Language Processing in Action: Understanding, analyzing, and generating text with Python*. New York: Manning Publications.

Mudding, A. A., & Karim, A. A. (2022). Analisis Sentimen Menggunakan Algoritma LSTM Pada Media Sosial. *Jurnal Publikasi Ilmu Komputer dan Multimedia, I*(3), 181-187. doi:https://doi.org/10.55606/jupikom.v1i3.517

Putra, J. W. (2020). *Pengenalan Konsep Pembelajaran Mesin dan Deep Learning.* Tokyo.

Scarpino, M. (2018). *TensorFlow for Dummies.* New Jersey: John Wiley & Sons, Inc.,.

Sharma, R., & Patel, M. (2018). Toxic Comment Classification Using Neural Networks and Machine Learning. *International Advanced Research Journal in Science, Engineering and Technology, 5*(9), 47-52.

Singh, P., & Manure, A. (2020). *Learn TensorFlow 2.0: Implement Machine Learning and Deep Learning Models With Python.* India: Apress.

Zaheri, S., Leath, J., & Stroud, D. (2020). Toxic Comment Classification. *SMU Data Science Review, 3*(1), 1-16.

Zhang, A., Lipton, Z., Li, M., & Smola, A. (2019). *Dive Into Deep Learning.* China: Creative Commons Attribution-ShareAlike 4.0 International Public License.