

LSTM and Bidirectional GRU Comparison for Text Classification

Hannan Asrawi^{1)*}, Ema Utami²⁾, and Ainul Yaqin³⁾

^{1,2,3)} Magister Teknik Informatika, Universitas Amikom Yogyakarta, Yogyakarta, Indonesia

¹⁾hannanasr@students.amikom.ac.id, ²⁾ema.u@amikom.ac.id, ³⁾ainulyaqin@amikom.ac.id

Submitted : Aug 17, 2023 | **Accepted** : Aug 23, 2023 | **Published** : Oct 1, 2023

Abstract: Although the phrases machine learning and AI are frequently used interchangeably and are frequently discussed together, they do not have the same meanings. While all artificial intelligence (AI) is machine learning, not all AI is machine learning, which is a key distinction. In the beginning, machine learning and natural language processing (NLP) are related since machine learning is frequently employed as a tool for NLP tasks. The advantage of NLP is that it can perform analysis, and examine a lot of data, including comments on social media accounts and hundreds of online customer evaluations. Text classification is essentially what needs to be done. This study compares Bidirectional GRU and LSTM as text classification algorithms using 20,000 newsgroup documents from 20 newsgroups from The UCI KDD Archive. After using the suggested model, we compare it to the long short-term memory and bidirectional GRU models for accuracy and validation. The results of the two comparisons show that the bidirectional GRU model performs better than the long short-term memory model. And this is a successful classification of text using a deep learning algorithm that uses a bidirectional GRU.

Keywords: Bidirectional GRU; LSTM; Machine learning; NLP; Text Classification

INTRODUCTION

The ability of robots to emulate intelligent human behavior is the general definition of the artificial intelligence subfield known as machine learning. Systems with artificial intelligence are utilized to carry out complicated tasks in a way that is comparable to how people solve problems. While all artificial intelligence (AI) is machine learning, not all AI is machine learning, which is a key distinction (Qaisar, 2020).

Machine learning is currently being used everywhere. Machine learning algorithms are used to make our interactions, online purchases, and social media usage efficient, streamlined, and secure. We are only beginning to scrape the surface of machine learning's capabilities as the field and the technology that supports it are developing quickly (Ma et al., 2019).

First, let's clarify how machine learning and natural language processing (NLP) relate to one another. Machine learning is frequently used as a tool for NLP tasks, therefore there is some overlap between machine learning and NLP. Statistical models, machine learning, deep learning, and computational linguistic rule-based modeling of human language are all combined in NLP (Tulu, 2022).

Consequently, NLP can be considered a subset of machine learning that enables computers to decipher, interpret, and produce human language. You ought to learn and apply NLP if you have a lot of written data and wish to discover new things. The advantage of NLP is that it allows for the analysis and close examination of a large amount of data, including comments on social media accounts and hundreds of customer evaluations posted online. Most importantly, text classification needs to be done (Jiao & Qu, 2019).

*name of corresponding author



This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

The process of classifying text into one or more groups in order to organize, arrange, and filter it according to any criteria is known as text classification. Text classification is utilized, for instance, in legal documents, medical studies and files, or even something as straightforward as product reviews. Since text/document data is far more prevalent than other types of data, it is crucial to find innovative ways to use it. (Cheng et al., 2021)

Deep learning techniques demonstrated their superiority in text classification by delivering cutting-edge outcomes on a number of common academic benchmark challenges. Classification using deep learning Deep learning is a technique that encourages computers to learn by doing, just like people do. Deep learning, a technology that enables self-driving cars to recognize stop signs and alert pedestrians from light poles, is a key element of the system(Fudholi et al., 2022)(Ji & Wu, 2020).

Long Short-Term Memory and Bidirectional Gated Recurrent Unit are the two deep learning algorithms that will thereafter be contrasted. A model featuring sequence processing and two GRUs is known as a bidirectional gated recurrent unit, or BiGRU. The input for one GRU is directed forward, while the input for the other is directed backward. Only input gates and forget gates are present in this bidirectional recurrent neural network. Long short-term memory (LSTM) is sometimes outperformed by a sort of recurrent neural network called a gated recurrent unit (GRU)(Cheng et al., 2021)

Recurrent neural networks of the Long Short-Term Memory (LSTM) type are able to recognize sequence relationships in sequence prediction issues. In complicated problem domains like machine translation, speech recognition, and other areas, this behavior is essential. A challenging subfield of deep learning is LSTM. It can be challenging to comprehend what an LSTM is and how concepts like bidirectional and sequence-to-sequence apply to this sector (Handayani et al., 2022)(Zulqarnain et al., 2019).

This research compares two deep learning models, LSTM and Bidirectional, in order to determine which model is superior for text categorization from the two models and to assess the performance of the two models. The trade-offs and designs of LSTM and GRU are very different. GRU has fewer gates and parameters than LSTM, which makes it easier and faster, but provides higher computational cost and risk of overfitting (Hamayel & Owda, 2021). LSTM has more gates and parameters than GRU, which makes it more robust and customizable. While the GRU has one hidden state that serves both purposes, which can limit its capacity, the LSTM has different cell states and outputs, allowing it to store and output different information. In addition, the sensitivity of LSTM and GRU can differ to hyperparameters such as sequence length, learning rate, or dropout rate (Yang et al., 2020).

LSTM and GRU can be regarded as effective RNN variations despite their differences because they share a few characteristics. Both make use of gates to control data flow and get around the issue with missing or exploding gradients. Both can recognize sequential patterns in data and spot enduring dependencies (Yurtsever, 2021). They can be layered into several levels to make the network deeper and more complicated. They can be combined with other neural network topologies, including convolutional neural networks (CNNs) or attention processes, to increase their performance(Yu et al., 2020).

Data from a number of sources, including 20 Newsgroup data, will be used in this study. These data pertain to 20,000 newsgroup documents that are distributed (nearly) uniformly among 20 different newsgroups. an excerpt from The UCI KDD Archive. Bidirectional GRU and LSTM are used to further categorize it. To determine which of the two deep learning algorithm models is the most effective in this situation, comparison is thought to be crucial.

The remaining parts of the essay are arranged as follows. The related study is discussed in Section 2, the dataset and the suggested approach are discussed in Section 3, the experimental findings are shown in Section 4, and the conclusions and next steps are discussed in Section 5.

LITERATURE REVIEW

Artificial neural networks have been effective in classifying documents in recent years. Defining a network for the entire document classification process without explicitly separating representation from classification is the first strategy in this trend. Applications for text classification as an electronic foundation for information retrieval, digital libraries, and other sectors are quite promising(Zulqarnain et al., 2019).

*name of corresponding author



Performance of text classification is frequently hampered by improving classification accuracy and semantic sensitivity of sparse data to context. In the research, a unified framework is proposed to examine the impacts of word embedding and Gated Recurrent Unit (GRU) for text classification on two included benchmark datasets (Google snippets and TREC). Experimentally, words with similar meanings are typically located close to one another in the embedding space. First, using the word embedding technique, the post's words are transformed into vectors. Once the contextual semantics between words have been extracted, the sentence's subsequent words are input to GRU. According to experimental findings, the suggested GRU model can successfully learn word usage in the context of the training text that has been presented. Training data quality and quantity have a big impact on performance (Zulqarnain et al., 2019).

The degree of community happiness is also gauged in other studies using text categorization. Because the review data is unstructured, it is not possible to determine user happiness solely by glancing at and examining the PLN Mobile review column in the Google Play store. A novel approach, sentiment analysis, is required to solve this issue. The goal of this study is to suggest a sentiment analysis architecture that would help deep learning algorithms like LSTM and GRU acquire crucial data. The suggested architecture includes word2vec as the word insertion and bidirectional GRU (BiGRU) as the attention mechanism. Important words are noted by the attention mechanism so that crucial information can be understood by the architecture. According to experimental findings, the suggested sentiment analysis architecture has a higher f1-score and accuracy (Chamidy et al., 2023).

LSTM and gated recurrent unit (GRU) networks are two well-known variations of artificial neural networks (RNNs) with long-term memory, and the study (Yang et al., 2020) investigated the performance differences of these two deep learning models, involving two dimensions. datasets for long- and short-text training, as well as for quantitative analysis of five metrics by taking into account two aspects of performance and processing power cost, GRU has a greater performance-cost ratio than LSTM, which is higher in accuracy ratio, recall ratio, and F1 ratio by 23.45%, 27.69%, and 26.95%, respectively.

The efficacy of LSTM networks in numerous real-world applications has led to extensive coverage of them in academic papers, technical blogs, and implementation manuals. The LSTM system equations and a full description of the LSTM system components. Our method and choice of notation for presenting the LSTM system stress simplicity, albeit being uncommon. Determine new options for LSTM system enrichment as part of the analysis, and then merge these additions into the Vanilla LSTM network to produce the most prevalent LSTM variety to yet

Creating methods for sentiment scoring hotel reviews, especially in Indonesian. The Long-Short Term Memory (LSTM) model and the Word2Vec model are both used in this study. Word2Vec architecture, Word2Vec vector dimension, Word2Vec evaluation method, pooling methodology, dropout value, and learning rate are among the Word2Vec and LSTM variables that were employed in the integration. The best result, with an accuracy of 85.96%, was discovered through experimental research using 2500 review texts as datasets (Muhammad et al., 2021).

Large amounts of data are being evaluated in the newly developed field of sentiment analysis in order to produce insightful conclusions about a certain subject. It is a powerful tool that may support organizations, businesses, and even consumers. In this system, text emotion recognition is crucial (Hermanto et al., 2021).

The sentiment of IMDb movie reviews is analyzed in the study using a Long Short-Term Memory (LSTM) classifier. Recurrent Neural Network (RNN) algorithm serves as its foundation. To enhance the performance of the post classification, the data is efficiently processed and partitioned. The classification performance's accuracy is investigated. With 89.9% classification accuracy, the findings are the best. This demonstrates the viability of incorporating the created method into current text-based sentiment analysis.

*name of corresponding author



METHOD

In our research, we compare bidirectional GRU and LSTM models with attention mechanisms and present a text analysis model for classifying text based on the aforementioned fundamental model. The following figure 1 depicts the model's structure:

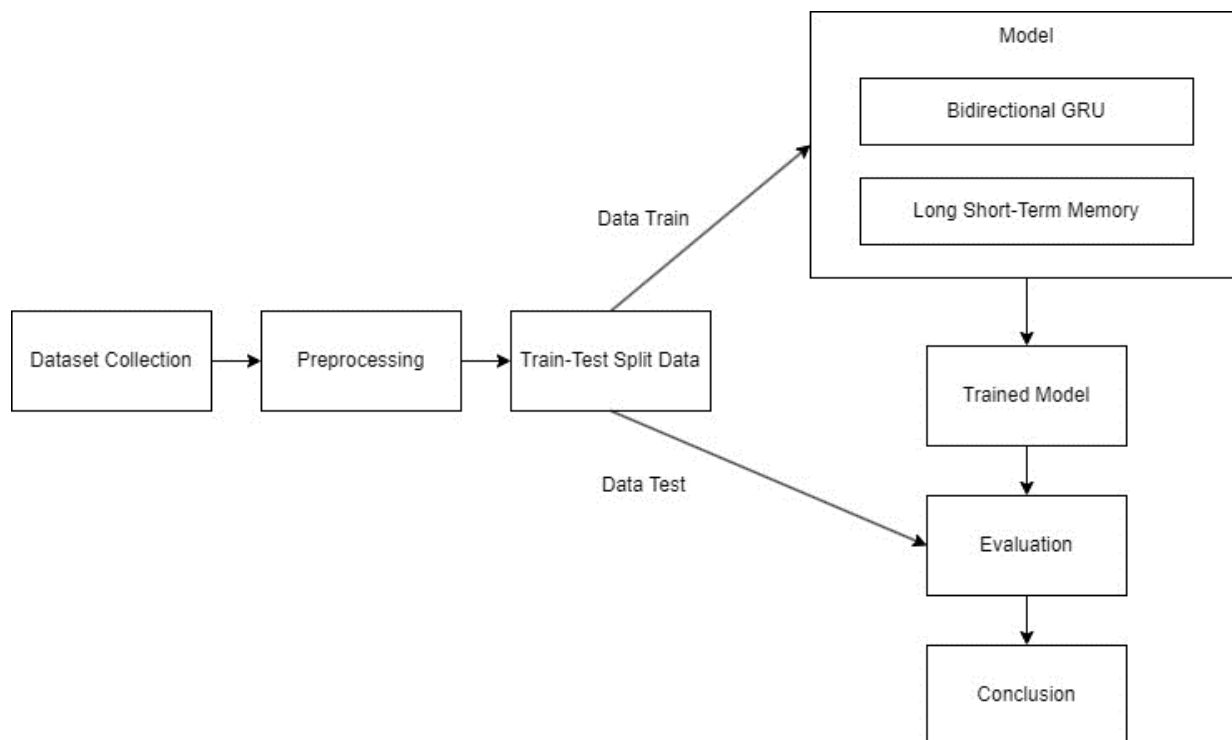


Figure 1. Model Diagram

Dataset and Pre-Processing

The newsgroup documents in this dataset were collected by The UCI KDD Archive in 1999a. A common dataset for machine learning studies in text applications, such as text classification and text clustering, is the collection of 20 newsgroups. The document is written in English, and a file called list.csv contains references to document_id numbers and the newsgroups that go along with them.

The data is initially pre-processed before moving on to the following step. In order to prepare the text data for feeding into the model, text preprocessing is a technique. Text data comprises noise in many different forms, including emotions, punctuation, and text in various circumstances. To do that, tokenization, lower case, stopword removal, and data cleaning techniques are used. It will also be categorized using Bidirectional GRU and LSTM (Jakaria et al., 2021)(Fransiska & Irham Gufroni, 2020).

The sklearn.datasets function was used to access data from 20 newsgroups in this study. The dataset was downloaded based on the recommended date variation of the dataset, and it included a point-in-time separation between the training set and the test set. The dataset size when downloaded is about 14 MB after compressed, while the training set is 52 MB, and the test set is 34 MB uncompressed. Next, build the Bidirectional GRU and LSTM deep learning model using the training dataset. after the training, testing will be carried out on the test data that has been prepared.

Bidirectional RNNs

Straight (past) and backward (future) input traversal is made possible by using bidirectional RNNs, also known as bidirectional RNNs. Two RNNs make up a BRNN, with one moving forward at the start of the data sequence and the other travelling backward at the end. (Salur & Aydin, 2020) The BRNN may contain a straightforward RNN, GRU, or LSTM network block. To facilitate the backward training procedure, bidirectional RNNs feature an additional hidden layer. The forward and backward concealed states are updated in the manner described below at any time t :

*name of corresponding author

$$A_t(Forward) = \phi(X_t * W_{XA}^{forward} + A_{t-1}(Forward) * W_{XA}^{forward} + b_A^{forward}) \quad (1)$$

$$A_t(Backward) = \phi(X_t * W_{XA}^{backward} + A_{t-1}(Backward) * W_{XA}^{backward} + b_A^{backward}) \quad (2)$$

It should be noted that ϕ represents the activation function, W the weight matrix, and b the bias. Combining $A_t(Forward)$ and $A_t(Backward)$, one may determine the concealed state at time t . Each particular concealed state's output is

$$O_t = H_t * W_{AY} + b_Y \quad (3)$$

Gated Recurrent Unit

At each time step, GRU will utilize a gating mechanism to update the network's hidden state in a limited fashion. Information entry and exit into the network are managed by the gating mechanism. The reset gate and the update gate are two of GRU's two gating systems. The update gate defines how much of the hidden state should be updated with fresh input, while the reset gate determines how much of the hidden state from the previous cycle should be forgotten. Based on the new hidden state, the GRU calculates its output. (Setiawan & Lestari, 2021).

The following equations are used to determine the GRU's reset gate, update gate, and hidden state:

Gates:

$$o_t = \sigma(W_{oht-1} + U_{oxt} + b_o) \quad (4)$$

$$i_t = \sigma(W_{iht-1} + U_{ixt} + b_i) \quad (5)$$

$$f_t = \sigma(W_{fht-1} + U_{fxt} + b_f) \quad (6)$$

States:

$$\tilde{s}_t = \sigma(W_{ht-1} + U_{xt} + b) \quad (7)$$

$$s_t = f_t \odot s_{t-1} + i_t \odot \tilde{s}_t \quad (8)$$

$$h_t = o_t \odot \sigma(s_t) \quad (9)$$

Long Short-Term Memory (LSTM)

Short-term memory networks are referred to as Long Short-Term Memory (LSTM) in the context of deep learning. It is a class of recurrent neural networks (RNN) that is particularly adept at learning long-term relationships in sequence prediction challenges.

LSTM is made up of several memory cells, which are many memory blocks, and four neural networks. An input gate, an output gate, a forget gate, and a cell make up a standard LSTM unit. Three gates regulate the cell's ability to receive and transmit information, and the cell can store data for any length of time. The LSTM method is well suited to classify, examine, and forecast time series of ambiguous duration. (Agustiniingsih et al., 2022)(Dicky Wahyu Hariyanto, 2020).

Information is stored in cells, while memory is controlled by gates. Three gates are present: The input gate selects the input value to be utilized to modify the memory. The sigmoid function decides whether a value of 0 or 1 is permitted to pass. Additionally, the tanh function delivers the input weight by ranking the significance of the data on a scale from -1 to 1.

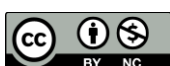
$$i_t = \sigma(W_i \cdot [h_t - 1, x_t] + b_i) \quad (10)$$

$$C_t = \tanh(W_c \cdot [h_t - 1, x_t] + b_C) \quad (11)$$

Forget Gate: It identifies the information that needs to be scrubbed from the block. A sigmoid function decides what it is. It examines the previous state (h_{t-1}) and the input content (X_t) for each number in the state of cell C_{t-1} before producing a number between 0 and 1 for that particular value.

$$f_t = \sigma(W_f \cdot [h_t - 1, x_t] + b_f) \quad (12)$$

*name of corresponding author



Input and block memory are utilized to determine the output of the output gate. Whether to let the value 0 or 1 through is decided by the sigmoid function. The tanh function also establishes which value can pass through 0 and 1. Additionally, the provided value is given weight using the tanh function, which rates its importance on a scale of -1 to 1 before multiplying it by the sigmoid output.

$$O_t = \sigma(W_o \cdot [h_t - 1, x_t] + b_o) \quad (13)$$

Hidden layer: influences the value in the following process. The value of this layer is derived from the output value multiplied by the value of the cell state or memory cell that has been triggered using the tangent function.

$$h_t = O_t * \tanh(C_t) \quad (14)$$

Evaluation

An epoch is an entire cycle of training a machine learning model using the entire training dataset; each training sample in the dataset is processed by the model during one epoch, and the weights and bounces of the dataset are calculated by the algorithm. An epoch also is the total number of iterations of all the training data in one cycle to train the machine learning model (Isnain et al., 2020). Each model's accuracy and loss outcomes are compared, and the comparison is then utilized as an evaluation to draw a conclusion.

RESULT

The outcomes of the data categorization tests that have been done will be explained in this section. The Jupyter Notebook application was used to perform this research using the Python programming language. From The UCI KDD Archive, 1999a, a dataset collection of 20 newsgroups was retrieved. But the `sklearn.datasets.fetch_20newsgroups` function was used as the source of the programming.

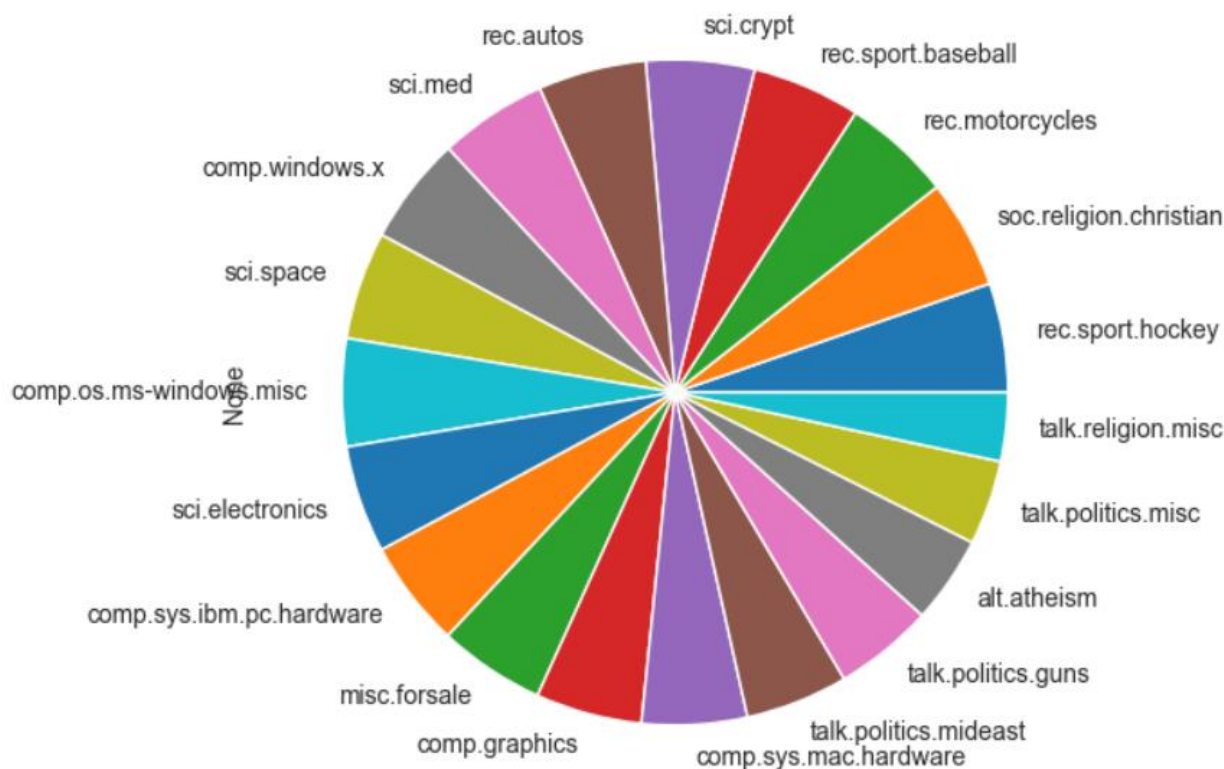


Figure 2. Target Class Distribution

*name of corresponding author



The first step in the process is data cleaning, which entails eliminating redundant words, looking for missing data, case folding (removing punctuation), changing all letters to lowercase, and changing all numbers on the page to lowercase [!"#\$%&'()*+,-./:;>?@[_`]]. Encoding separates the text into tiny units called tokens for additional analysis, while stopwords elimination eliminates frequently used words that are deemed to have little value.

The sklearn.datasets method is used in this study to access data from 20 newsgroups. There are roughly 18,000 posts on 20 subjects in the 20 newsgroup dataset. One subset of these posts is designated for testing (or performance evaluation), whereas the other is designated for training or development. Based on messages posted before and after a specific date, the training and testing sets were divided.

A deep learning algorithm's ability to stack numerous layers of models makes it convenient to use. As shown in Figure 3, the first layer in a multi-layer bidirectional GRU model is embedding, which needs integer input. The output of the embedding layer is a set of 2D vectors, each of which represents a word from the lexicon.

Instead of dropping individual elements, the second layer, SpatialDropout1D, drops the entire 1D feature map. Regular dropout will not regularize the activation and will only lead to a decrease in the effective learning rate if neighboring frames within the feature map are significantly correlated, as is typically the case in the initial convolution layer. Instead, SpatialDropout1D should be utilized since it will enhance feature map independence.

In Figure 3, the Bidirectional layer will duplicate the passed RNN layer and flip its go_backwards field to process the inputs in the opposite order. By default, the forward layer output and the backward layer output will be concatenated to create the output of the bidirectional GRU. The output of the final layer, which is dense, is the dot product of the input tensor and the weight matrix, or kernel. The output value is then added to the bias vector value, which is followed by an element-by-element activation process.

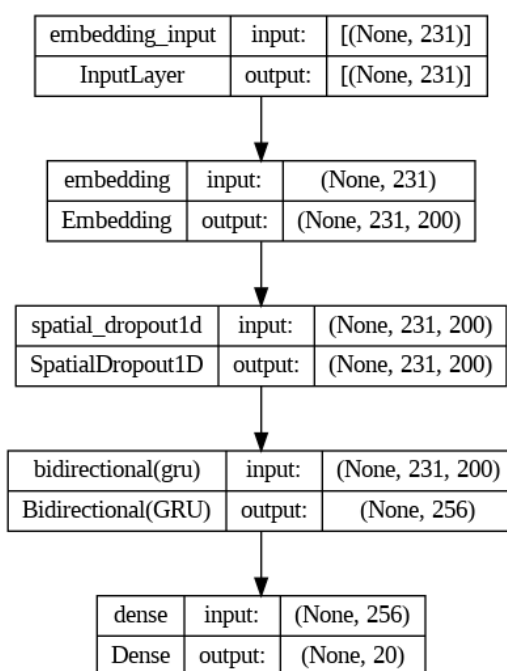


Figure 3. An overview of the bidirectional GRU model

The LSTM model in Figure 4 uses a sequential paradigm similar to a standard layer stack with each layer having precisely one input tensor and one output tensor, similar to the Bidirectional GRU model. The Bidirectional GRU model's embedding is the same. This network's LSTM layer is made up of a hidden layer with four LSTM blocks or neurons, a visible layer with one input, and an output layer that predicts a single value. The LSTM blocks make use of the default sigmoid activation function. LSTMs

*name of corresponding author

always have a 3D array as their input layer. Depending on the argument, the LSTM can produce either a 2D array or a 3D array, however in this instance, the output is a 2D array.

The output of the dense next layer is the dot product of the weight matrix, also known as the kernel, and the input tensor. The softmax function is also used by the activation layer to transform the value vector into a probability distribution. The output vector's components, which vary from 0 to 1, add up to 1. Every vector is dealt with separately. The axis argument tells the function which axis of the input to use. Because the outcome can be understood as a probability distribution, Softmax is frequently employed as the activation for a classification network's last layer. The LSTM model is illustrated in Figure 4 below.

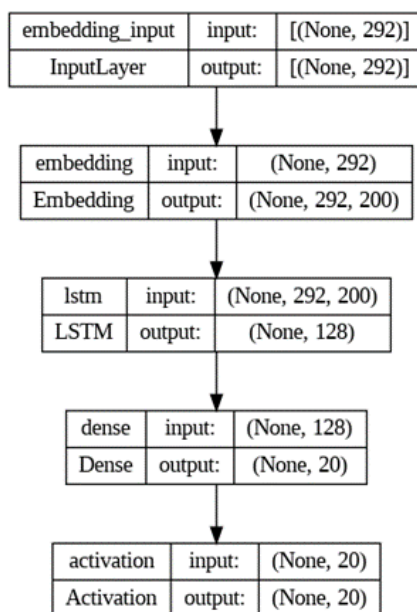


Figure 4. An overview of the Long Short-Term Memory model

The bidirectional GRU model is the first to be discussed, and Figure 5 shows the following outcomes.

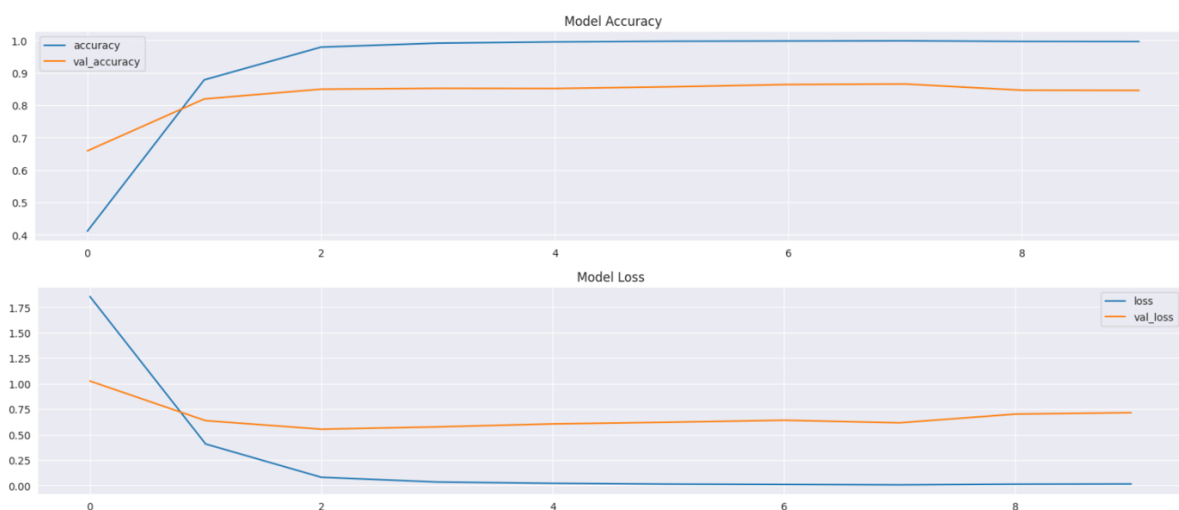


Figure 5. Accuracy and loss in Bidirectional GRU Model

The accuracy value is 0.9987 and the validation accuracy value is 0.7257 based on the graph in Figure 5 from the results of 11 epochs. Figure 5 shows a comparison of epoch loss values. The loss is calculated as 0.0082, and the validation loss is calculated as 1.2189 and the loss model. The accuracy calculation for this experiment yielded results with an epoch value of 11 times, 99% accuracy, and a loss of 0.8%.

*name of corresponding author



The experiment's findings, which were based on the long short-term memory hypothesis, are shown in Figure 6.

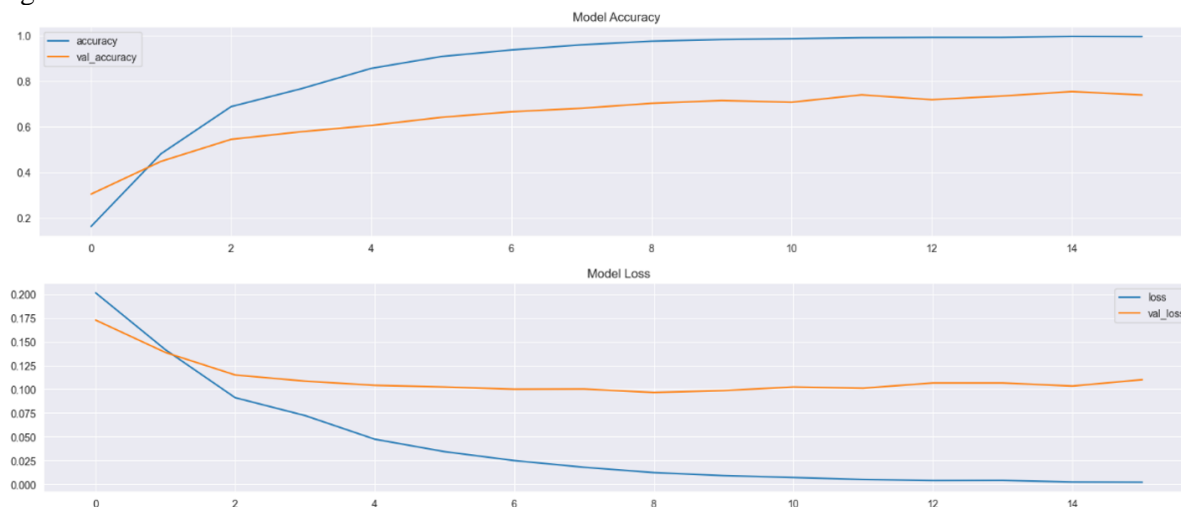


Figure 6. Accuracy and loss in LSTM Model

The accuracy value is 0.9956 and the validation accuracy value is 0.7395 based on the graph in Figure 6 from the results of 16 epochs. Figure 6 shows a comparison of the values of epoch loss. The outcome is a loss of 0.0022 and a validation loss of 0.1099. compares favorably to the loss model. In this experiment, the accuracy calculation yielded data with an epoch value of 16 times, 99% accuracy, and loss of 0.22%. The differences between the two models used above for comparison can be noticed from these results

DISCUSSION

The implemented Bidirectional GRU model is structured as follows. In particular in the initial convolutional layer, the spatial dropout 1D layer uses a 2x2 filter size to account for the correlation of nearby pixels. In essence, you want to stop pixels from adapting to their surroundings across the feature map and force them to learn as if there are no other feature maps. SpatialDropout2D encourages feature map independence precisely in this way. GRU layer that is bidirectional If you use a GRU with 128 units for a bidirectional input layer, the GRU output is a 2D dense layer with 20 units and a sigmoid activation function.

The optimizer "adam," loss function "binary_crossentropy," and metric "accuracy" are used to do compilation after the model is generated. This indicates that the model will train using the Adam approach, compute the loss using categorical crossentropy, and display the accuracy metric throughout training and evaluation.

The results of the experiment provide the following explanation. The model has 11 training iterations, or epochs. Information such as the loss function, accuracy on training data, loss function, and accuracy on validation data (in this case, val_loss and val_accuracy) are displayed at each epoch as described in the results section of tables 1 and 2. The test data were evaluated using the model at the end of the most recent epoch (the 11th epoch), and the accuracy and loss function were 0.9987 and 0.0082, respectively.

We will then go over the outcomes of the used LSTM model, which has the following structure:

The LSTM layer is used in a sequential model that processes a sequence of integers, embeds each integer into a vector, and then processes the sequence of vectors. LSTM layer that is 128 units in size. Each recurrent layer in Keras includes two arguments relating to dropout with a recurrent dropout of 0.1: dropout, a float defining the dropout rate for the layer's input units, and recurrent_dropout, defining the dropout rate of the recurrent units. Classification was carried out using a Dense layer with 20 units and a softmax activation function as an addition.

The results of the experiment provide the following explanation:

*name of corresponding author



The model has 16 training iterations, called epochs. The experiment is carried out similarly to the previous model in that information about the loss function, accuracy on training data, loss function, and accuracy on validation data (in this case, val_loss and val_accuracy) is displayed at each epoch. This information is explained in the results section of tables 3 and 4. A model evaluation on the test data was done at the end of the 16th epoch, and the results showed an accuracy of 0.9956 and a loss function of 0.0022.

The limits of this study only consider the period outcomes, which are then compared between the two models to reach a judgment. Performance evaluations in binary classification problems and multiclass classification problems, however, can be added as suggestions for additional research.

CONCLUSION

In this text classification that employs data from The UCI KDD Archive, which is a 20-newsgroup dataset made up of roughly 18,000 newsgroup postings on 20 subjects, we apply bidirectional GRU with respect to text data and compare it with Long Short-Term Memory. It is categorized using bidirectional GRU to acquire experimental findings using data from 10 epochs, which produced an accuracy value of 0.9965 and a validation accuracy value of 0.8456. Hence the outcome of a loss of 0.0144 and a loss of 0.7144 during validation.

When compared to the outcomes of the Long Short-Term Memory model, the outcomes of 16 epochs had accuracy values of 0.9956 and 0.7395 for validation. Loss of 0.0022 and validation loss of 0.1099 produce comparison of epoch loss values.

Following the two comparisons, it can be said that the long short-term memory model is less efficient than the bidirectional GRU model. And this is a successful text classification using a bidirectional GRU deep learning system.

It is recommended for future research using other data, and it is expected to use machine learning model evaluation measurements, it is very important to ensure that the machine learning model has good performance.

ACKNOWLEDGMENT

We appreciate the assistance of the Master of Informatics Engineering teachers at Amikom University in Yogyakarta as well as the supervisors, tutors, and all of our friends and colleagues.

REFERENCES

- Agustiningsih, K. K., Utami, E., Muhammad, O., & Aلسyaibani, A. (2022). *Sentiment Analysis of COVID-19 Vaccines in Indonesia on Twitter Using Pre-Trained and Self-Training Word Embeddings*. 1, 39–46.
- Chamidy, T., Informatika, M., Islam, U., Maulana, N., Ibrahim, M., & Mechanism, A. (2023). *Bidirectional GRU dengan Attention Mechanism pada Analisis Sentimen PLN Mobile*. 22(2), 358–372.
- Cheng, Y., Sun, H., Chen, H., Li, M., Cai, Y., Cai, Z., & Huang, J. (2021). Sentiment Analysis Using Multi-Head Attention Capsules with Multi-Channel CNN and Bidirectional GRU. *IEEE Access*, 9, 60383–60395. <https://doi.org/10.1109/ACCESS.2021.3073988>
- Dicky Wahyu Hariyanto, W. M. (2020). *Analisis Sentimen Pada Media Sosial Twitter*. 12(1), 79–86.
- Fransiska, S., & Irham Gufroni, A. (2020). Sentiment Analysis Provider by.U on Google Play Store Reviews with TF-IDF and Support Vector Machine (SVM) Method. *Scientific Journal of Informatics*, 7(2), 2407–7658. <http://journal.unnes.ac.id/nju/index.php/sji>
- Fudholi, D. H., Zahra, A., & Nayoan, R. A. N. (2022). A Study on Visual Understanding Image Captioning using Different Word Embeddings and CNN-Based Feature Extractions. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, 4(1), 91–98. <https://doi.org/10.22219/kinetik.v7i1.1394>
- Hamayel, M. J., & Owda, A. Y. (2021). A Novel Cryptocurrency Price Prediction Model Using GRU, LSTM and bi-LSTM Machine Learning Algorithms. *Ai*, 2(4), 477–496. <https://doi.org/10.3390/ai2040030>
- Handayani, S. F., Pratiwi, R. W., Dairoh, D., & Af'idah, D. I. (2022). Analisis Sentimen pada Data

*name of corresponding author



- Ulusan Twitter dengan Long-Short Term Memory. *JTERA (Jurnal Teknologi Rekayasa)*, 7(1), 39. <https://doi.org/10.31544/jtera.v7.i1.2022.39-46>
- Hermanto, D. T., Setyanto, A., & Luthfi, E. T. (2021). Algoritma LSTM-CNN untuk Binary Klasifikasi dengan Word2vec pada Media Online. *Creative Information Technology Journal*, 8(1), 64. <https://doi.org/10.24076/citec.2021v8i1.264>
- Isnain, A. R., Sihabuddin, A., & Suyanto, Y. (2020). Bidirectional Long Short Term Memory Method and Word2vec Extraction Approach for Hate Speech Detection. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 14(2), 169. <https://doi.org/10.22146/ijccs.51743>
- Jakaria, A., Mu'minah, S., Riana, D., & Hadianti, S. (2021). Klasifikasi Varietas Buah Kiwi dengan Metode Convolutional Neural Networks Menggunakan Keras. *Jurnal Media Informatika Budidarma*, 5(4), 1309. <https://doi.org/10.30865/mib.v5i4.3166>
- Ji, C., & Wu, H. (2020). Cascade architecture with rhetoric long short-term memory for complex sentence sentiment analysis. *Neurocomputing*, 405, 161–172. <https://doi.org/10.1016/j.neucom.2020.04.055>
- Jiao, Y., & Qu, Q. X. (2019). A proposal for Kansei knowledge extraction method based on natural language processing technology and online product reviews. *Computers in Industry*, 108, 1–11. <https://doi.org/10.1016/j.compind.2019.02.011>
- Ma, S., Tong, L., Ye, F., Xiao, J., Bénard, P., & Chahine, R. (2019). Hydrogen purification layered bed optimization based on artificial neural network prediction of breakthrough curves. *International Journal of Hydrogen Energy*, 44(11), 5324–5333. <https://doi.org/10.1016/j.ijhydene.2018.12.142>
- Muhammad, P. F., Kusumaningrum, R., & Wibowo, A. (2021). Sentiment Analysis Using Word2vec and Long Short-Term Memory (LSTM) for Indonesian Hotel Reviews. *Procedia Computer Science*, 179(2020), 728–735. <https://doi.org/10.1016/j.procs.2021.01.061>
- Qaisar, S. M. (2020). Sentiment Analysis of IMDb Movie Reviews Using Long Short-Term Memory. *2020 2nd International Conference on Computer and Information Sciences, ICCIS 2020*, 12–15. <https://doi.org/10.1109/ICCIS49240.2020.9257657>
- Salur, M. U., & Aydin, I. (2020). A Novel Hybrid Deep Learning Model for Sentiment Classification. *IEEE Access*, 8, 58080–58093. <https://doi.org/10.1109/ACCESS.2020.2982538>
- Setiawan, E. I., & Lestari, I. (2021). Stance Classification Pada Berita Berbahasa Indonesia Berbasis Bidirectional LSTM. *Journal of Intelligent System and Computation*, 3(1), 41–48. <https://doi.org/10.52985/insyst.v3i1.148>
- Tulu, C. N. (2022). Experimental Comparison of Pre-Trained Word Embedding Vectors of Word2Vec, Glove, FastText for Word Level Semantic Text Similarity Measurement in Turkish. *Advances in Science and Technology Research Journal*, 16(4), 147–156. <https://doi.org/10.12913/22998624/152453>
- Yang, S., Yu, X., & Zhou, Y. (2020). LSTM and GRU Neural Network Performance Comparison Study: Taking Yelp Review Dataset as an Example. *Proceedings - 2020 International Workshop on Electronic Communication and Artificial Intelligence, IWECAL 2020*, 98–101. <https://doi.org/10.1109/IWECAL50956.2020.00027>
- Yu, W., Yi, M., Huang, X., Yi, X., & Yuan, Q. (2020). Make it directly: Event extraction based on tree-LSTM and Bi-GRU. *IEEE Access*, 8, 14344–14354. <https://doi.org/10.1109/ACCESS.2020.2965964>
- Yurtsever, M. (2021). Gold Price Forecasting Using LSTM, Bi-LSTM and GRU. *European Journal of Science and Technology*, 31(31), 341–347. <https://doi.org/10.31590/ejosat.959405>
- Zulqarnain, M., Ghazali, R., Ghouse, M. G., & Mushtaq, M. F. (2019). Efficient processing of GRU based on word embedding for text classification. *International Journal on Informatics Visualization*, 3(4), 377–383. <https://doi.org/10.30630/joiv.3.4.289>

*name of corresponding author



This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.