International Journal of Applied Science and Engineering

Improving service quality through classifying chatbot messages based on natural language processing: A bidirectional long short-term memory network model

Christine Dewi [1*], Calvin Satya Adi Kristiantoro [1], Henoch Juli Christanto [2*], Dalianus Riantama [3]

[1] *Department of Information Technology, Satya Wacana Christian University, Salatiga City, 50711, Indonesia*
[2] *Department of Information System, Atma Jaya Catholic University of Indonesia, Jakarta, 12930, Indonesia*
[3] *Management Department, BINUS Business School Undergraduate Program, Bina Nusantara University, Jakarta, 11480, Indonesia*

ABSTRACT

The rapid development of the times and the influence of globalization have enormously changed human life. One of the affected fields is service, and machines are gradually replacing services. Not without reason, the number of bad assessments of services is one of the factors why machines begin to replace the role of humans. The existence of machines also makes it easier for companies to provide services and help cut costs for labor. The machine used in this research was a chatbot. Chatbot is a computer program designed to simulate conversations between humans. The long short-term memory network (LSTM) algorithm was implemented on the chatbot with a natural language processing (NLP) approach in this research. Our experiment was carried out using the NLP approach, where the results were used in the data training process using the bidirectional LSTM algorithm to produce a chatbot model. Next, after evaluating the model, our proposed method outperformed other models in the experiment. Bidirectional LSTM had 98.09% accuracy, 98.23% precision, 98.29% recall, and 98.25% f1 score.

Keywords: Chatbot, LSTM, NLP.

OPEN ACCESS

Received: June 8, 2023
Revised: August 9, 2023
Accepted: January 29, 2024

Corresponding Author:
Henoch Juli Christanto
henoch.christanto@atmajaya.ac.id

Publisher:
Chaoyang University of Technology
ISSN: 1727-2394 (Print)
ISSN: 1727-7841 (Online)

1. INTRODUCTION

In the era of the Industrial Revolution 4.0, services are gradually being replaced by machines. This is not without reason; this is done to be able to help human performance and improve service quality in various service sectors. Eight cross-sectional study articles from Asia, Europe, Africa, and North America were analyzed and concluded that there is a need to improve service quality by 2.61 times in terms of both management and administration (Rokhmatun et al., 2023). Other research such as that conducted by Fida et al. (2020) on a Bank in the Sultanate of Oman states that there must be an increase in service to a very satisfactory level due to the significant relationship between service quality, customer satisfaction, and customer loyalty.

To enhance service quality, implementing a chatbot on web platforms or applications is a viable option. A chatbot is a computer program designed for human-like conversations, equipped with artificial intelligence and natural language processing to intelligently respond to user queries (Luo et al., 2022). Increasingly utilized, especially in the business sector, chatbots serve as substitutes for customer service, contributing to positive experiences for users in terms of system, service, and information quality (Zhou et al., 2020; Jenneboer et al., 2022).

In this study, the recurrent neural networks and long short-term memory (RNN-LSTM) algorithm are employed to develop the chatbot. The long short-term memory method facilitates natural language processing for effective communication between users and computers. Research findings, such as those by Anki et al. (2021) and Anki and Bustamam (2021), indicate the high accuracy of chatbots using LSTM, reaching around 0.994869. The long short-term memory method has been quite widely used in research on making chatbots because it can receive various inputs and issue outputs in the form of data sequences which are the result of the development of the recurrent neural network (RNN) method. LSTM has an advantage over RNN because it can read long sequences (Calimeri et al., 2019).

RNN has a problem called vanishing gradient or loss of gradient effectiveness. RNNs cannot remember historical data over a long period (Wei et al., 2023). With LSTM, the vanishing gradient problem can be handled. LSTM is also able to extract contextual information more quickly from the past and the present without having to remove the meaning of the context (Wang et al., 2020).

The main contributions of this research are summarized as follows: (1) we create a chatbot by implementing the LSTM algorithm using the TensorFlow module to facilitate company and society performances in providing services. The anticipated outcome of utilizing the chatbot was to enhance the service quality of corporations or facilitate the public in acquiring information from relevant institutions. (2) We evaluate and classify chatbot messages with bidirectional LSTM. Afterward, we provide a detailed analysis of each model.

The remainder of the paper is organized as follows. Section 2 presents an overview of the materials and methods used. Section 3 presents our results and discussion, and section 4 describes our conclusions and future research goals.

2. MATERIALS AND METHODS

2.1 Chatbot

Chatbot is a program or application that uses artificial intelligence to be able to process input in the form of text and provide answers or responses from the processed input. More simply, a chatbot is a computer program designed to simulate conversations with human users, especially over the Internet. Chatbot will automatically retrieve keywords from the most similar input patterns and then provide the most appropriate reply from the stored textual database (Suhaili et al., 2021). Like a human conversation partner, a chatbot helps provide answers to questions. Until now, chatbots can be found in several social media networks such as Telegram, Facebook Messenger, WeChat, Twitter, and Instagram (Dhandayuthapani, 2022). Chatbots are very helpful for companies in providing effective services for 24 hours and of course in saving labor (Hardi et al., 2022). Chatbots are also known as smart bots, interactive agents, digital assistants, or artificial conversation entities (Adamopoulou and Moussiades, 2020).

2.2 Natural Language Processing (NLP)

Natural language processing (NLP) is a branch of artificial intelligence that focuses on the interaction between machines and humans using human language (DeSouza et al., 2021). Computer machines will learn how to understand and process human language to be processed and provide output to users (Khurana et al., 2023). The purpose of NLP is to build a system that knows natural language both from the structure of the sentence, the meaning of the word, and the meaning of a sentence to be able to communicate in a context (Yuniar and Purnomo, 2019). In general, NLP has the task of breaking down input sentences into smaller pieces, and then understanding how those pieces can relate to create meaning (Chowdhary, 2020). In this research, NLP is made using a Keras library with a TensorFlow platform and a Python programming language.

Keras is a high-level application programming interface (API) for constructing and testing deep learning models (Chicho and Sallow, 2021). Keras lets users use or extract model features without constructing their own. Keras can adjust deep learning model-building parameters to meet user needs, making it versatile (Grattarola and Alippi, 2021). Therefore, Keras is ideal for this research. TensorFlow runs deep learning algorithms properly. Keras compensates for TensorFlow's weaknesses when architecture differs, or a fast optimizer is needed in research.

2.3 Long Short-Term Memory (LSTM)

LSTM is a type of neural network architecture that is the result of the development of RNN or recurrent neural network to be able to overcome the dependency problem (Oruh et al., 2022). When compared to RNN, LSTM has a gate to be able to eliminate vanishing and exploding gradient problems found in RNN (Long and Zeng, 2022). LSTM works by using memory cells to keep information from the past and then combining it with current data to make a prediction (Hu et al., 2020). LSTM is the most appropriate model for use in processing sequential sentences or text categorization and machine translation processes.

In one LSTM neuron, some layers interact with each other, namely a tanh layer and a sigmoid layer. The sigmoid layer functions to return values in the range of zero and one. The tanh layer serves to return values in the negative range of one-to-one. The LSTM structure is divided into 4 components including forget gate, input gate, cell state, and output gate. More details about LSTM can be seen in Fig. 1.

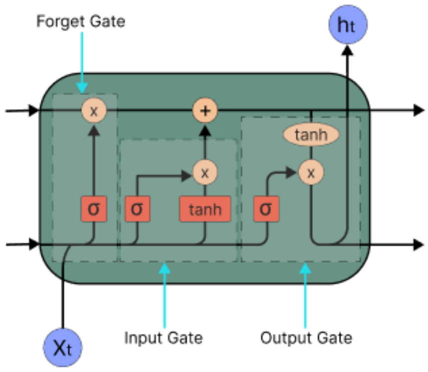

Fig. 1. LSTM architecture

Forget gate is a gate that has a role in forgetting information that is considered irrelevant and not needed by the system. This information is obtained from previous layers and information from the current layer. The value calculation will be done using Equation (1), if the result gets a value of 0 then the information will be discarded but if it gets a value of 1 then the information will be stored (Lin et al., 2020). With this gate, LSTM will only present a collection of actual and complete information as needed.

$$f_t = \sigma_g\left(W_f x_t + U_f h_{t-1} + b_f\right) \tag{1}$$

f_t: Forget gate
$W_f\, x_t$: The input weight
U_f : The weight of the cyclic connection
h: The output vector
t: The time
b_f: The partial weight
σ_g: Sigmoid activation function

Input gate is the second gate that has a role in ensuring that the information obtained and used is appropriate and accurate. The input gate will provide additional information to the result of the forget gate selection. This gate returns the value to 0 or 1 (Chusna et al., 2022). One of the reasons why LSTM is better than RNN is also because of the input gate. The input gate formula can be seen in Equation (2).

$$i_t = \sigma_g\left(W_t x_t + U_t h_{t-1} + b_t\right) \tag{2}$$

i_t : Input gate

Cell state has the most important role, namely controlling the flow of information. Cell state can store information that is needed from previous units. Cell State can store information that is needed from previous units. The information is passed to the tanh layer to change the value range to be between -1 to 1 (Setiawan et al., 2020) and produce a candidate cell state as in Equation (3). The next stage is doing pointwise multiplication operations as in Equation (4) to get the desired output on the cell state (Hastomo et al., 2022).

$$\tilde{c}_t = \sigma_h\left(W_c x_t + U_c h_{t-1} + b_c\right) \tag{3}$$

$$c_t = f_t \; c_{t-1} + i_t \; \tilde{c}_t \tag{4}$$

$\tilde{c}$: Intermediate cell state
σ_h: tanh function
C_t: Cell state vector

Output gate is the last gate to be able to produce information that has been filtered both actually and completely. To be able to determine the output gate, you can use Equation (5). In the next step, the sigmoid layer will determine the cell state for output, the result of the cell state is then processed in the tanh layer and then multiplied by the sigmoid gate as stated by Lin et al. (2021). This can be seen in Equation (6). The output gate can be the last part for actual and complete information or the first part to be processed in the next cell's input gate.

$$o_t = \sigma_g\left(W_o x_t + U_o h_{t-1} + b_o\right) \tag{5}$$

$$h_t = o_t \sigma_h\left(c_t\right) \tag{6}$$

o_t: The activation vector of the output gate
h_t: Output vector

2.4 Research Workflow

The workflow of this research can be seen in Fig. 2, which will explain how this program is created. Several software is needed to be able to build a chatbot using the LSTM algorithm. The first is Python as a programming language with Keras as a library and TensorFlow as a machine learning framework to be able to create a chatbot. Furthermore, there is Google Collaboratory as a tool for writing code, data pre-processing, data modelling, training, and validation. Then there is also a lot of data that is put together in an after with JavaScript Object Notation (JSON) format.

2.4.1 Dataset

Dataset is a database that resides in memory. Datasets can be interpreted as objects that represent data and their relationships in memory. More clearly, a dataset is a collection of structured data that is ready to be used to obtain new data. In this research, the dataset was created independently, by searching for frequently asked keywords on the internet. The topics (tags) determined were also inspired by features in similar applications that have been deployed on mobile applications. Next phase the data set is stored in a file with the JSON format. The dataset consists of several parts including:

- Intents is a group of all data inputted and the output from the input results. In dataset modelling, intents contain tags, inputs, and responses.
- Tags is a place where a group of data in the form of similar text is grouped into one group. Several different inputs can get the same output or response if they are classified into the same tags.
- Input is a place where a pattern of data can be inputted by the user in this case in the form of text.
- Response is a place that contains patterns of output data that will be automatically sent to the user for the results of the input that has been given.

In creating the dataset, the result is a dataset consisting of 13 tags/categories, 1050 sentences of input, and 13 sentences of response. Below is shown one of the first tags/categories along with the input and response in JSON format.

2.4.2 Parsing the Data

Parsing data is the process of taking data in one format and converting it to another format. Parsing will identify the main parts of the sentence such as object, subject, and others (Togatorop et al., 2021). Parsing will map the sentence into a parse tree and will look useful when it takes several data forms in one data unit.

In this research, the dataset is imported first for data pre-processing. The first stage of data pre-processing is downloading stop words from the natural language toolkit (NLTK) library which is an application of NLP. Stop word itself is a word that often appears such as conjunctions "and", "or", “but”, "will", and others. These words are ignored to make the speed and performance of NLP much better. In addition, data pre-processing will also remove numbers, symbols, and punctuation marks.

The final stage of data processing is to make everything lowercase. After data pre-processing, the dataset is moved into a list data type and then converted into a data frame. The data frame itself is a two-dimensional table that contains data presented in rows and columns. Data frames are very useful to facilitate researchers in processing raw data. The data frame can be seen in Table 1.

2.4.3 Train-test Split

Train-test split is one of the methods that can be used to evaluate the performance of machine learning models. This model evaluation method divides the dataset into two parts training data and testing data with a certain proportion. The train data is used to fit the machine learning model, while the test data is used to evaluate the model fit results. This method is used to provide accurate prediction results regarding the performance of data that has never been trained. In this research, the input data is put into sentences while the tags data is put into labels. Sentences and labels will hold source and target data for training and testing purposes. In addition, the test size parameter is also determined as 0.8, which means that 80% of the data from the dataset is used for testing data purposes. The result is 840 sentences for training, 840 labels for training, 210 sentences for validation, and 210 labels for validation.

2.4.4 Tokenizing

The data that has been separated through the train test split process is then forwarded to the tokenizing process. Tokenizing is one of the NLP methods that will cut a text or sentence into smaller units called tokens. Tokenizing uses spaces as delimiters so that it can produce tokens in the form of a word. Based on the trials that have been run, about 336 unique words are obtained from the data. The 336 words are then converted into numeric form.

2.4.5 Sequences, Truncating and Padding

At this stage, the tokenized data is converted into sequences. Sequence is an array or list that contains tokens of each word in a sentence. These tokens are in numeric form obtained from the tokenizing process. However, not all sentence structures have the same length. In this research, a length of 6 words is made so that it will have a similar size and can later be processed in the machine learning model. If the number of sequences in the sentence is more than the specified length, then truncating will be done. Truncating is reducing the sequence to the limit of the length. Likewise, if the number of sequences in the sentence is less than the specified length, padding will be done. Padding is the addition of sequences up to the max length limit. Padding will add index 0 up to the max. length limit. In this research, the addition and subtraction of sequences are done at the end because the truncating and padding parameters are filled with the value 'post'.

2.4.6 Encoding

Like the tokenizing process, the encoding stage converts the tags into numeric form. If tokenizing is changing the input, then encoding is changing the tags. If the tags have been converted into numeric form, they can be used directly for the training and validation process.

2.4.7 Embedding

In this stage, natural language is classified using global vectors for word representations (GloVe). GloVe is an unsupervised learning algorithm for obtaining vector representations of words (word embeddings). This process will allow the machine learning model to understand the meaning of each word or number/vector representation. If several words have similar vectors, they will be grouped into the same group. In this research, approximately 400,000-word vectors were found.

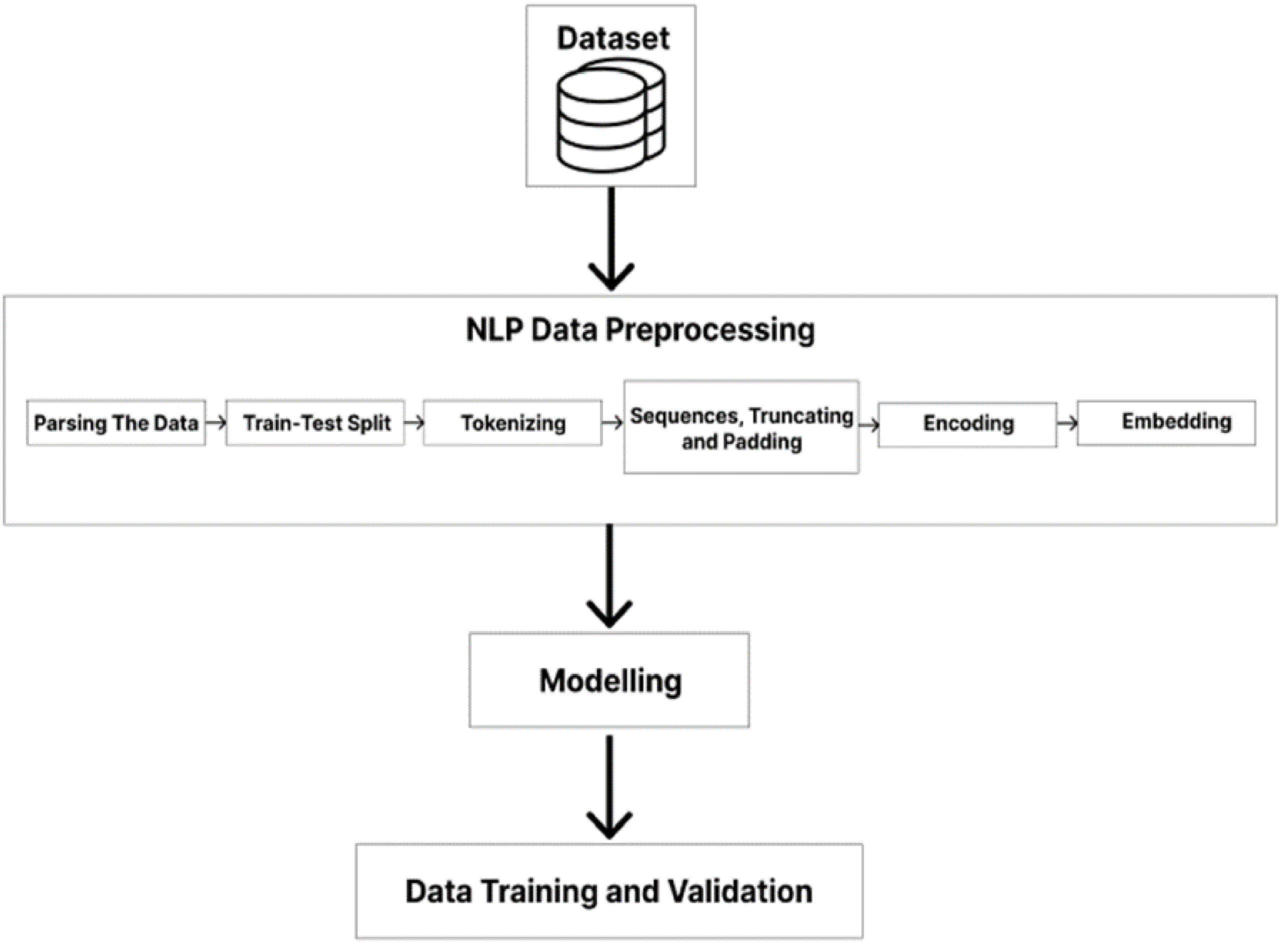

Fig. 2. The workflow of this research

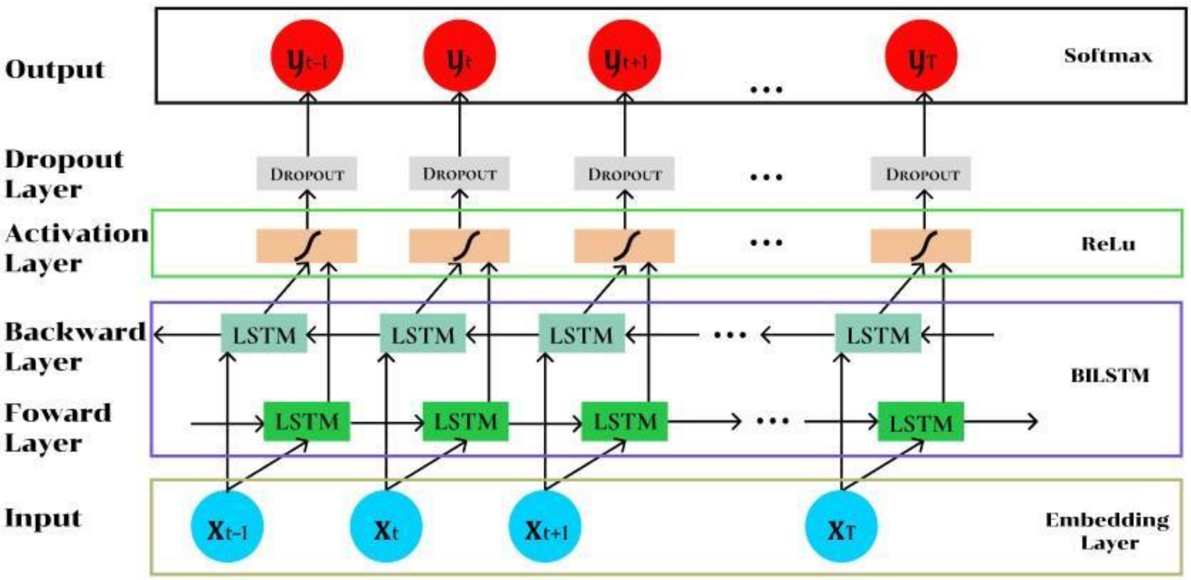

Fig. 3. Model design

Table 1. Converting results to a data frame

No.	Inputs	Tags
0	hello	greeting
1	halo	greeting
2	hola	greeting
3	hey	greeting
4	hoy	greeting
…	…	…
1045	merci	closing
1046	many thanks	closing
1047	say thanks	closing
1048	can't thank you enough	closing
1049	million thanks	closing

2.4.8 Modelling

At this stage, model building is carried out. This stage is still related to the training and validation stages. Further, there are 5 layers designed for the chatbot development process. More clearly, the design model can be seen in Fig. 3. The first layer is the embedding layer. This layer is commonly known as the input layer which contains data that has passed the text preprocessing stage. The second layer is the bidirectional LSTM Layer. Bidirectional LSTM is a development of the LSTM model where two layers process in the opposite direction. This model is very good at recognizing sentence patterns because it processes each word sequentially. The bottom layer will move forward, so it will understand and process from the first word to the last word. The upper layer will move backward, so it will understand and process from the last word to the first word. This layer allows us to learn past and future information for each input sequence. The third layer is the hidden layer, where the rectified linear unit (ReLu) activation function is implemented. The ReLu function has the advantage of taking a randomly initiated network. If there is an element with a negative value, then ReLu will change the value to 0. The formula and graph of the ReLu activation function can be seen in Equation (7) and Fig. 4.

The fourth layer is the dropout layer which serves to ignore neurons randomly to prevent overfitting. Overfitting itself is a condition where machine learning has accurate predictions but only for training data, not for new data.

The last layer is the output layer, where the SoftMax activation function is implemented. The SoftMax function calculates the probability of several events. In the case of machine learning, this function will calculate the probability of each label it guesses. The advantage of this activation function is that the output value ranges from 0 to 1. If each SoftMax function result is summed up, it will have a value of 1. The formula and graph of the SoftMax activation function can be seen in Equation (8) and Fig. 5.

These five layers are compiled and then get a summary as shown in Table 2. Our experiment employs five layers including the embedding, bidirectional, dense 4, dropout, and dense 5 layer. The total parameter is 49425 with 0 non-trainable parameters. The embedding layer is commonly the first layer in a neural network used for NLP. A bidirectional layer performs computations on an input sequence in both the forward and backward directions. Dense layers are conventional neural network layers in which each neuron is coupled to every neuron in the preceding and succeeding layers. Dropout is a technique used to regularize neural networks to mitigate the problem of overfitting. An additional dense layer could be used either for the final output or for an intermediate representation.

3. RESULTS AND DISCUSSION

3.1 Data Training and Validation

The next process is the data training process. This stage is carried out repeatedly until it gets a higher level of accuracy and a lower loss rate. In this research, the data training process was carried out with an epoch of 150. From the results of the training data, the testing accuracy value is approximately 0.9670 and the validation value is 0.9810. The loss value in testing is also very small which has a value of around 0.1649 and a loss in validation of around 0.1180. This proves that the model for chatbot design is good enough to be implemented. For more details, it is shown in Table 3 which is the last 10 epoch data. Figs. 6 and 7 also show a comparison graph of training and validation accuracy and loss.

3.2 Model Testing

At this stage, the model performance is tested by comparing it with other algorithms. In general, this test uses the same dataset and hyperparameters but different algorithms. The algorithms compared are bidirectional LSTM, LSTM, simple RNN, and gated recurrent unit (GRU). Four values from the confusion matrix were used to check if these values were classified correctly. The confusion matrix is a summary of the prediction results that classify a new object into a class (misclassification) from a classification job (Chen et al., 2020). The four values of the confusion matrix are true positive (TP), true negative (TN), false positive (FP), and false negative (FN). True positive is the number of "yes" in the actual situation when the model evaluation also stated "yes". (Dewi et al., 2023). True negative is the inverse of true positive where the number of samples that are not recognized (Chen et al., 2022). False positive (FP) is a misclassified positive sample and false negative is the negative sample of misclassification or the number of misrecognized samples. (Tai et al., 2020; Dewi and Chen, 2022).

There are several evaluation methods used as indicators of model evaluation including precision, recall, and accuracy. Precision refers to the value that is truly positive on all data. Recall refers to the value determined as true by the recall will be true. Accuracy refers to the percentage of all the data that is correctly evaluated as true. For the precision, accuracy, recall, and f1- score calculation formula can be seen in Equations (9)–(12).

$$f(x) = \{0\ for\ x \leq 0\ x\ for\ x > 0 \quad (7)$$

$$f(x) = (o, x) \quad (8)$$

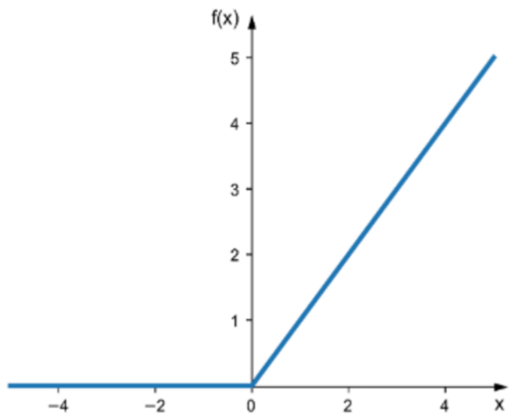

Fig. 4. ReLu activation function graph

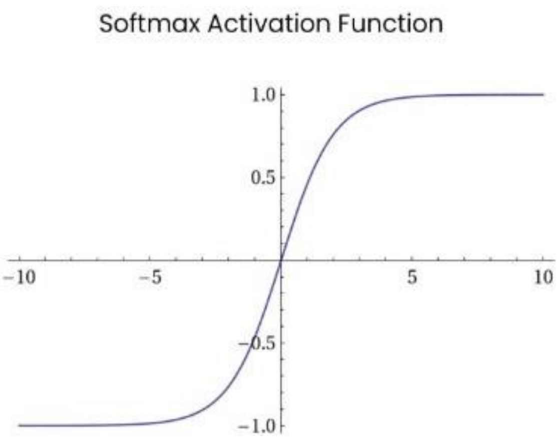

Fig. 5. SoftMax activation function graph

$$Precision = \frac{TP}{TP+FP} \quad (9)$$

$$Accuracy = \frac{TP}{TP+FP+FN+TN} \quad (10)$$

$$Recall = \frac{TP}{TP+FN} \quad (11)$$

$$F1 - score = 2 \times \frac{Precision\ x\ Recall}{Precision + Recall} \quad (12)$$

After evaluating the model, the results are shown in Fig. 8. As shown in Fig. 8, bidirectional LSTM gets the best results in terms of accuracy, precision, recall, and f1-score. Bidirectional LSTM has an accuracy of 98.09% while LSTM is only 92.85%, simple RNN is 73.81% and GRU is only 94.28%. Likewise, the precision value of bidirectional LSTM has a value of 98.23% while LSTM is only 93.31%, simple RNN is 74.56% and GRU is only 94.83%. For the recall value, bidirectional LSTM has a value of 98.29%, LSTM of 93.43%, simple RNN of 75.07%, and GRU of 94.76%. The f1-score value is the same bidirectional LSTM still outperforms others with a value of 98.25% while LSTM is only 93.36%, simple RNN is 74.82% and GRU is only 94.62%.

The experiment result performance can be increased by adding data to the dataset. To prove this, at least a total of 7 tags/categories, 772 sentences of input, and 7 responses were added to the dataset. The results of the addition will be seen in Fig. 9 and Table 5. In Fig. 8, the value will significantly increase as the data increases. Indirectly this also proves that this model is very competent to be applied to more complex situations.

As can be seen, the bidirectional LSTM algorithm outperforms the other algorithms. In more detail, the classification report and confusion matrix of the Bidirectional LSTM model are shown in Table 4 and Fig. 10. By using the confusion matrix method, the system tested the sentence that was inputted, and then the system also classified the sentence into the respective types of tags. It can be seen in Fig. 10, that the results of the errors we made are very minimal and almost non-existent for some tags. This proves that the model we made is good enough to be implemented. We also tried to do Blackbox testing, by inputting some random sentences and the results were as desired as shown in Fig. 11.

Table 2. Summary of the compiled model

Layer (type)	Output shape	Param #
embedding_2 (embedding)	(None, 6, 100)	33700
bidirectional_2 (bidirectional)	(None, 32)	14976
dense_4 (dense)	(None, 16)	528
dropout (dropout)	(None,16)	0
dense_5 (dense)	(None, 13)	221

Total params: 49,425
Trainable params: 49,425
Non-trainable params: 0

Table 3. Last 10 epoch data

Epoch	Loss	Accuracy	Val_loss	Val_accuracy
141	0.1673	0.9655	0.1264	0.9762
142	0.1593	0.9702	0.1238	0.9762
143	0.1511	0.9798	0.1246	0.9762
144	0.1872	0.9571	0.1239	0.9762
145	0.1604	0.9714	0.1215	0.9762
146	0.1529	0.9667	0.1173	0.9810
147	0.1537	0.9714	0.1180	0.9810
148	0.1577	0.9619	0.1195	0.9810
149	0.1617	0.9583	0.1165	0.9762
150	0.1649	0.9679	0.1180	0.9810

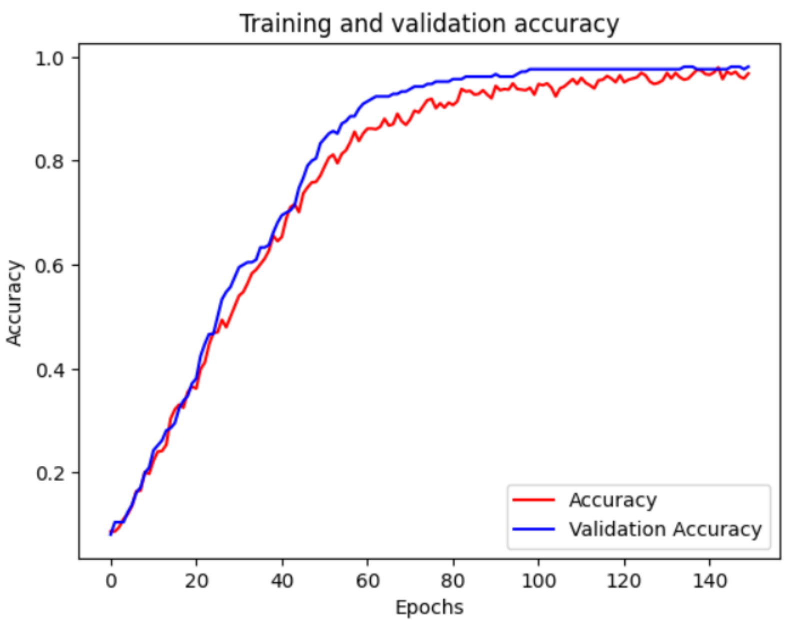

Fig. 6. Comparison graph of training and validation accuracy

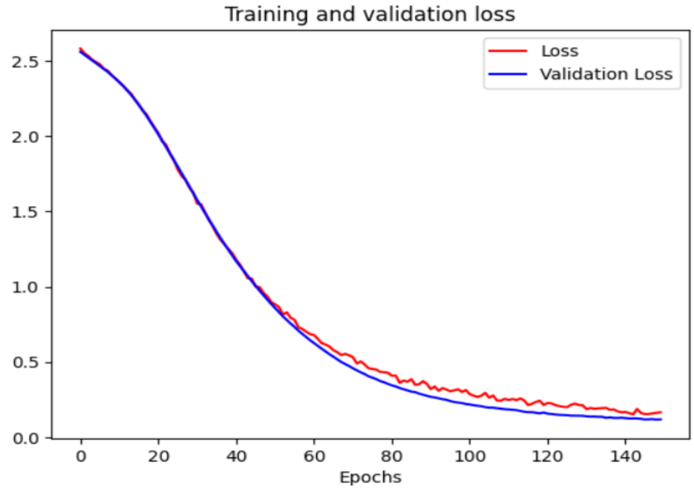

Fig. 7. Comparison graph of training and validation loss

Furthermore, the results of our study were compared with research conducted by Alghifari et al. (2022). With their research, we compared models from the same algorithm, namely bidirectional LSTM. From the model they have made, we try to add an activation function. In addition, we also made a larger dataset as suggested. The results show that our model has higher accuracy. We also compared the results using the confusion matrix method and concluded that our model is much better with 98.09% accuracy, 98.23% precision, 98.29% recall, and 98.25% f1-score. We also compared our research with that of Wintoro et al. (2022). In their research, the accuracy they got was very high but the loss value they had was also very high. This is inversely proportional to our research which has high accuracy but has a low loss value. In addition, they did not evaluate the results using the confusion matrix method or the like. They only test using Blackbox testing so it is questionable whether the accuracy results are valid or manipulated. This is questionable for their model because usually the higher the accuracy result, the lower the loss value.

From a series of studies that we have made, the result of our proposed method is to use a combination of the Bidirectional LSTM algorithm and activation function. In addition, the creation of large datasets also contributes to affecting the accuracy obtained. The larger the dataset made, the greater the accuracy obtained. Our research obtained 98.09% accuracy, 98.23% precision, 98.29% recall, and 98.25% f1 score. This proves that the model we have created is good to be implemented into a chatbot.

4. CONCLUSION

In this research, the implementations of LSTM on chatbot are very suitable for use. From the results of the training data, the accuracy of testing is approximately 96.70% and validation is 98.10%. For the loss value on the results testing is also very small which has a value of around 0.1649 and a loss on validation around 0.1180. In addition, an evaluation method using a confusion matrix is also carried out. The bidirectional LSTM obtained the best results in terms of accuracy, precision, recall, and f1-score when compared to LSTM, simple RNN, and GRU. Bidirectional LSTM has a 98.09% accuracy, 98.23% precision, 98.29% recall, and 98.25% f1 score. This proves that the model for chatbot design is good enough to be implemented.

Our findings give managers advice on how to strategically create a chatbot model from a practical aspect. Chatbot adds a new layer of support to the service quality dimension by guaranteeing that personalized service is available to meet customer needs whenever and wherever they may be, in response to the increased demand for the personalization of service delivery to meet specific consumer needs (Chung et al., 2020). Chatbots and other new technological tools enable businesses to simultaneously exceed consumer expectations, achieve corporate objectives, and generate value. It is important to note that value is not only a component of products/services alone (Sands et al., 2021), but can also be a component of the services offered by the digital agent involved in the service interaction. In the future, we will explore other classification methods to improve the performance of chatbots such as transfer learning and deep learning. We will also try to implement the chatbot we created in various service sectors while analyzing the chatbot's performance for the further development process.

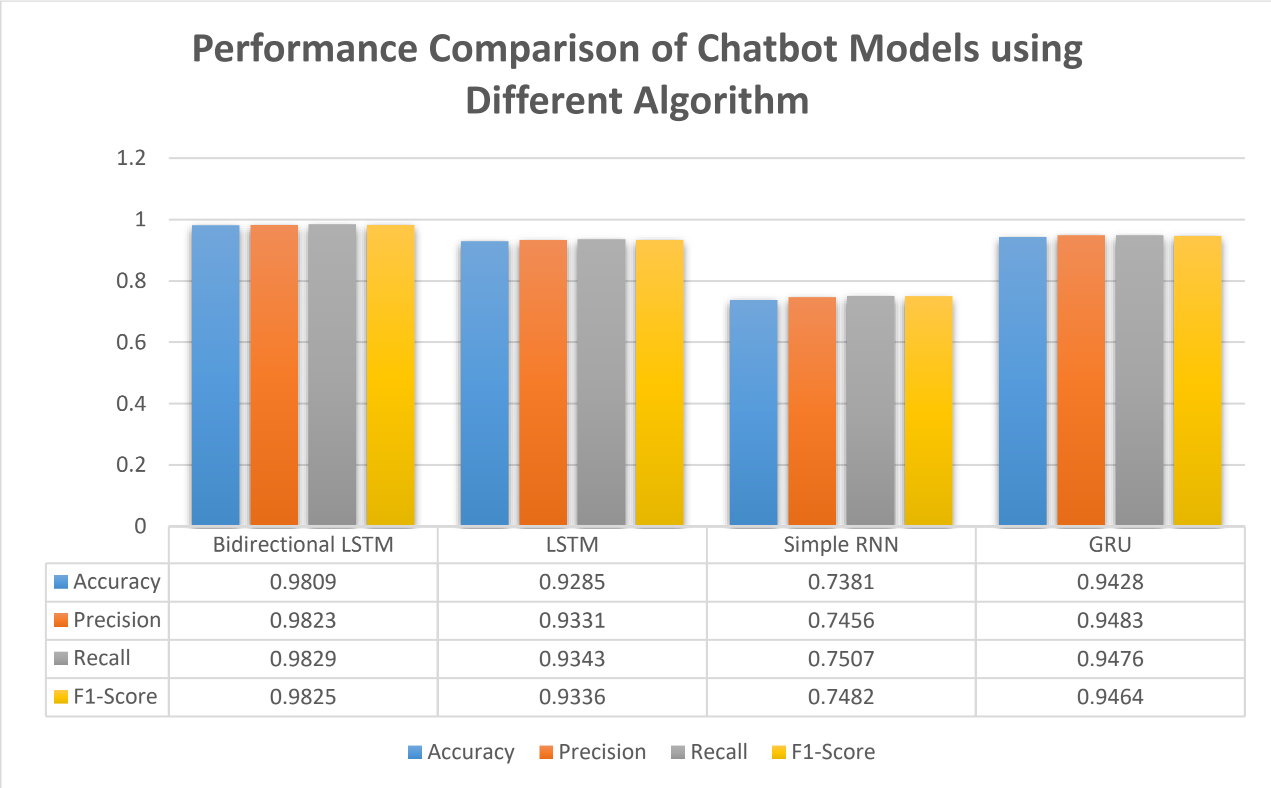

Fig. 8. Chart comparison of performance chatbot models using different algorithms

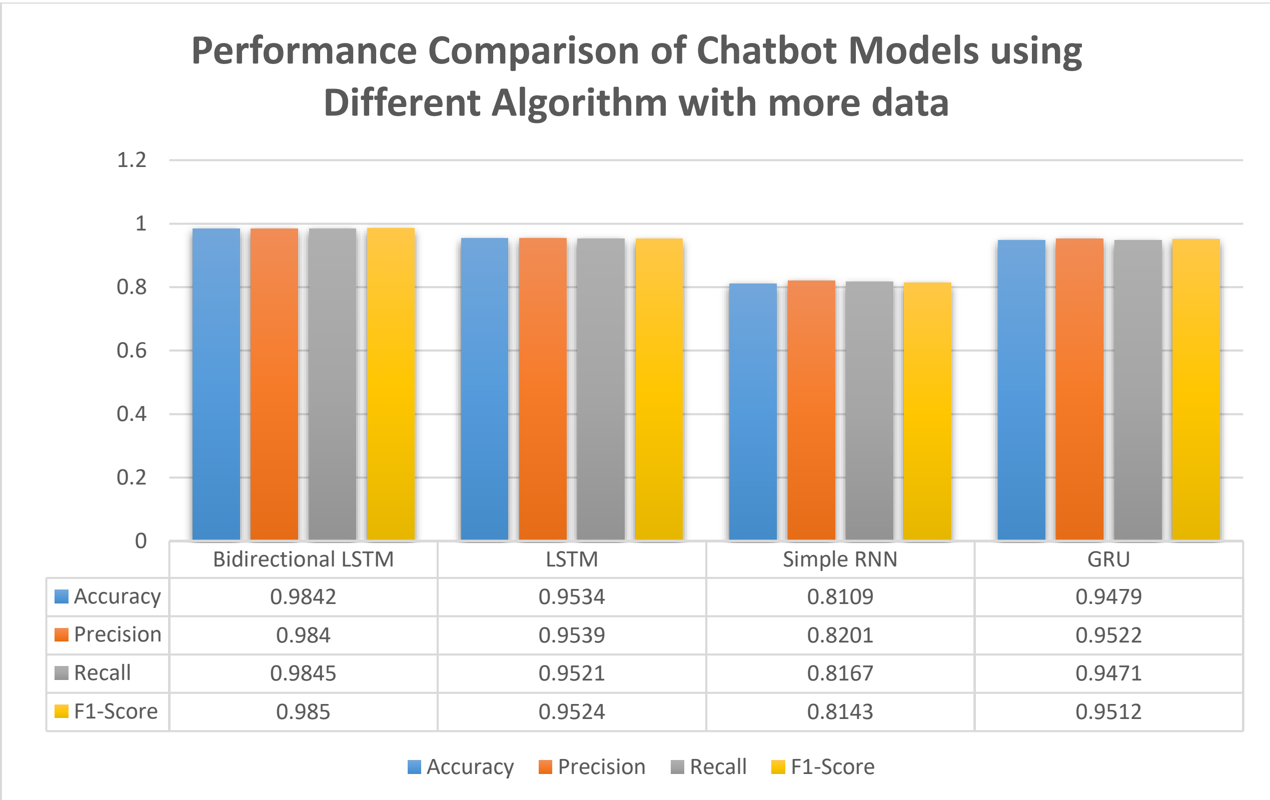

Fig. 9. Chart comparison of performance chatbot models using different algorithms with more data

Table 4. Classification report of the bidirectional LSTM model

Type	Precision	Recall	F1-score	support
Greeting	1.00	1.00	1.00	12
general payment info	0.94	1.00	0.97	15
booking flight	1.00	0.94	0.97	18
booking hotels	1.00	1.00	1.00	17
hotels recommendation	0.94	1.00	0.97	17
restaurant recommendation	0.95	1.00	0.97	18
hotels reschedule	0.94	1.00	0.97	16
flight reschedule	1.00	0.90	0.95	20
hotel refund	1.00	1.00	1.00	18
boarding ticket refund	1.00	1.00	1.00	14
call refund	1.00	0.93	0.97	15
travel credit	1.00	1.00	1.00	16
Closing	1.00	1.00	1.00	14
Accuracy			0.98	210
macro avg.	0.98	0.98	0.98	210
weighted avg.	0.98	0.98	0.98	210

Table 5. Classification report of the bidirectional LSTM model with more data

Type	Precision	Recall	F1-score	support
Greeting	1.00	1.00	1.00	11
general payment info	1.00	0.97	0.98	18
booking flight	0.98	0.94	0.97	21
booking hotels	1.00	1.00	1.00	14
hotels recommendation	0.94	0.97	0.96	10
restaurant recommendation	1.00	1.00	1.00	18
hotels reschedule	0.97	0.94	0.98	18
flight reschedule	0.96	1.00	0.98	31
hotel refund	0.94	1.00	0.98	14
boarding ticket refund	1.00	1.00	1.00	16
call refund	1.00	1.00	1.00	18
travel credit	0.94	0.98	0.96	17
closing	0.98	1.00	0.98	21
visa	1.00	0.98	0.98	23
promo	1.00	1	1	22
event	0.97	0.96	0.97	18
point	1.00	1.00	1.00	13
travel insurance	1.00	1.00	1.00	17
transportation	1.00	0.97	0.98	21
trip recommendation	1.00	0.98	0.98	24
Accuracy			0.98	365
macro avg.	0.98	0.98	0.98	365
weighted avg.	0.98	0.98	0.98	365

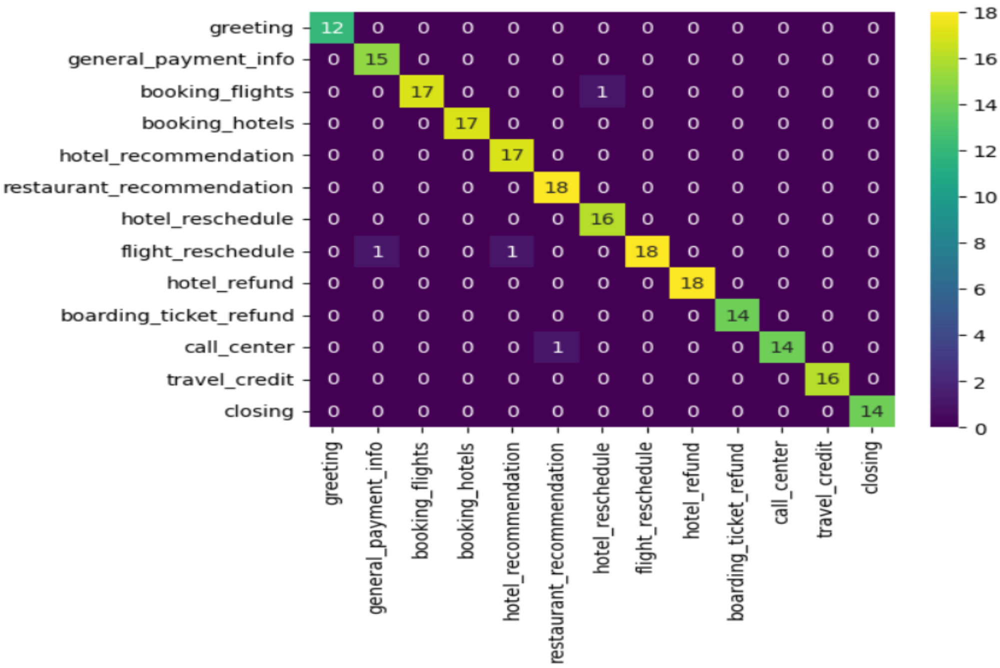

Fig. 10. Confusion matrix of the bidirectional LSTM model

Fig. 11. The chatbot's predicted response

ACKNOWLEDGMENT

This paper is funded by DIREKTORAT RISET DAN PENGABDIAN MASYARAKAT Satya Wacana Christian University, Indonesia.

REFERENCES

Adamopoulou, E., Moussiades, L. 2020. An overview of chatbot technology. IFIP Advances in Information and Communication Technology, 584, 373–383.

Alghifari, R.D., Edi, M., Firmansyah, L. 2022. Implementasi bidirectional LSTM untuk analisis sentimen terhadap layanan grab Indonesia. Jurnal Manajemen Informatika, 12(2), 89–99. (In Indonesian)

Anki, P., Bustamanm A. 2021. Measuring the accuracy of LSTM and BiLSTM models in the application of artificial intelligence by applying a chatbot program. Indonesian Journal of Electrical Engineering and Computer Science, 23(1), 197–205.

Anki, P., Bustamam, A., Al-Ash, S.H., Sarwinda, D. 2021. Intelligent chatbot adapted from question and answer system using RNN-LSTM model. Journal of Physics: Conference Series, 1844 012001.

Calimeri, F., Cauteruccio, F., Cinelli, L., Marzullo, A., Stamile, C., Terracina, G., Durand-Dubief, F., Sappey-Marinier, D. 2019. A logic-based framework leveraging neural networks for studying the evolution of neurological disorders. Theory and Practice of Logic Programming.

Chicho, T.B., Sallow, B.A. 2021. A comprehensive survey of deep learning models based on Keras framework. Journal of Soft Computing and Data Mining, 2(2), 49–62.

Chen, R.-C., Dewi, C., Huang, S.-W., Caraka, E.R. 2020. Selecting critical features for data classification based on machine learning methods. Journal of Big Data, 7(52), 1–26.

Chen, R.-C., Dewi, C., Jiang, X., Yu, H. 2022. Adjusting eye aspect ratio for strong eye blink detection based on facial landmarks. PeerJ Computer Science, 8, e943.

Chowdhary, K.R. 2020. Natural language processing. In: Fundamentals of artificial intelligence. Springer, New Delhi. pp 603–649.

Chung, M., Ko, E., Joung, H., Kim, S. J. 2020. Chatbot e-service and customer satisfaction regarding luxury brands. Journal of Business Research, 117, 587–595.

Chusna, L.N., Tari, R.M.D., Khumaidi, A. 2022. Konfigurasi hyperparameter long short-term memory untuk optimalisasi prediksi penjualan. Faktor Exacta, 15(4), 290–300.

DeSouza, D.D., Robin, J., Gumus, M., Yeung, A. 2021. Natural language processing as an emerging tool to detect late-life depression. Frontiers in Psychiatry, 12, 719125.

Dewi, C., Chen R.-C. 2022. Combination of resnet and spatial pyramid pooling for musical instrument identification. Cybernetics and Information Technologies, 22(1), 104–116.

Dewi, C., Chen, S.P.A., Christanto, J.H. 2023. Deep learning for highly accurate hand recognition based on Yolov7 model. Big Data and Cognitive Computing, 7(1), 53.

Dhandayuthapani, V. 2022. A proposed cognitive framework model for a student support chatbot in a higher education institution. International Journal of Advanced Networking and Applications, 14(2), 5390–5395.

Fida, A.B., Ahmed, U., Al-Balushi, Y., Singh, D. 2020. Impact of service quality on customer loyalty and customer satisfaction in islamic banks in the sultanate of oman. SAGE Open, 10(2), 1–10.

Grattarola, D., Alippi, C. 2021. Graph neural networks in TensorFlow and Keras with Spektral. IEEE Computational Intelligence Magazine, 16(1), 99–106.

Hardi, R., Pee, C.N.A., Abdullah, B.L.H.M., Pitogo, A.V., Pribadi, S.A., Rusdi, F.J. 2022. Academic smart chatbot to support emerging artificial intelligence conversation. 2022 International Conference of Science and Information Technology in Smart Administration (ICSINTESA). Denpasar, Bali, Indonesia, 194–199.

Hastomo, W., Aini, N., Karno, B.S.A., Rere, R.M.L. 2022. Machine learning methods for predicting manure management emissions. Jurnal Nasional Teknik Elektro dan Teknologi Informasi, 11(2), 131–139.

Hu, J., Wang, X., Zhang, Y., Zhang, D., Zhang, M., Xue, J. 2020. Time series prediction method based on variant LSTM recurrent neural network. Neural Processing Letters, 52, 1485–1500.

Jenneboer, L., Herrando, C., Constantinides, E. 2022. The impact of chatbots on customer loyalty: A systematic literature review. Journal of Theoretical and Applied Electronic Commerce Research, 17(1), 212–229.

Khurana, D., Koli, A., Khatter, K., Singh, S. 2023. Natural language processing: State of the art, current trends and challenges. Multimedia Tools and Application, 82, 3713–3744.

Lin, C.-B., Dong, Z., Kuan, Wei.-K., Huang, Y.F. 2021. A framework for fall detection based on openpose skeleton and LSTM/GRU models. Applied Sciences 11(1), 329.

Long, L., Zeng, X. 2022. Recurrent neural network. In: Beginning deep learning with TensorFlow. Apress, Berkeley, CA.

Luo, B., Lau, K.Y.R., Li, C., Si, Y.-W. 2022. A critical review of state-of-the-art chatbot designs and applications. WIREs Data Mining and Knowledge Discovery, 12(1), e1434.

Oruh, J., Viriri, S., Adegun, A. 2022. Long short-term memory recurrent neural network for automatic speech recognition. IEEE Access, 10, 30069–30079.

Rokhmatun, N.P., Maesaroh, S., Putri, R.R.I., Salmah, S., Kurnianingsih, A.S., Fauziah, N.A., Zamani, A., Atmojo T.J., Widiyanto, A. 2023. The relationship between hospital administration and quality of services: Meta-analysis. Journal of Health Policy and Management, 8(1),

39–47.
Sands, S., Ferraro, C., Campbell, C., Tsao, H.Y. 2021. Managing the human–chatbot divide: How service scripts influence service experience. Journal of Service Management, 32(2), 246–264.
Setiawan, I.E. Ferdianto, A., Santoso, J., Kristian, Y., Gunawan, Sumpeno, S., Purnomo, H.M. 2020. Indonesian stance analysis of healthcare news using sentence embedding based on LSTM. Jurnal Nasional Teknik Elektro dan Teknologi Informasi, 9(1), 8–17.
Suhaili, M.S., Salim, N., Jambli, N.M. 2021. Service chatbots: A systematic review. Expert Systems with Applications, 184, 115461.
Tai, S.-K., Dewi, C., Chen, R.-C, Liu, Y.-T., Jiang, X., Yu, H. 2020. Deep learning for traffic sign recognition based on spatial pyramid pooling with scale analysis. Applied Sciences, 10(19), 6997.
Togatorop, R.P., Simanjuntak, P.R., Manurung, B.S., Silalahi, C.M. 2021. Generating entity relationship diagram from requirement specification using natural language processing for indonesian language. Jurnal Komputer dan Informatika, 9(2), 196–206.
Wang, Q., Peng, R.-Q., Wang, J.-Q., Li, Z., Qu, H.-B. 2020. NEWLSTM: An optimized long short-term memory language model for sequence prediction. IEEE Access, 8, 65395–65401.
Wei, L., Guo, D., Chen, Z., Yang, J., Feng, T. 2023. Forecasting short-term passenger flow of subway stations based on the temporal pattern attention mechanism and the long short-term memory network. ISPRS International Journal of Geo-Information, 12(1), 25.
Wintoro, B.P., Hermawan, H., Muda, A.M. Mulyani, Y. 2022. Implementation of long short-term memory in the unila informatics engineering academic information chatbot. EXPERT: Jurnal Manajemen Sistem Informasi dan Teknologi, 12(1), 68–75.
Yuniar, E., Purnomo H. 2019. Implementasi chatbot “ALITTA” asisten virtual dari balittas sebagai pusat informasi di balittas. Antivirus: Jurnal Ilmiah Teknik Informatika, 13(1), 24–35. (In Indonesian)
Zhou, L., Gao, J., Li, D., Shum, H.-Y. 2020. The design and implementation of Xiaoice, an empathetic social chatbot. Computational Linguistics, 46(1), 53–93.