ISSN 1816-6075 (Print), 1818-0523 (Online)
Journal of System and Management Sciences
Vol. 13 (2023) No.3, pp. 505-516
DOI:10.33168/JSMS.2023.0334

Improving Chatbot Performance using Hybrid Deep Learning Approach

Palaninchamy Naveen [1], Su-Cheng Haw [1], Devakumaran Nadthan [1] ,Saravana Kumar Ramamoorthy [2]

[1] Faculty of Computing and Informatics, Multimedia University, 63100, Cyberjaya, Malaysia

[2] Tech Mahindra ICT Services, Global Solution Center, Cyberjaya, Malaysia

p.naveen@mmu.edu.my

Abstract. A chatbot is a computer program that is implemented to communicate with clients via natural language. Suruhanjaya Syarikat Malaysia (SSM) is a Malaysian statutory authority that governs businesses and companies and it would benefit from implementing a chatbot for a multitude of reasons, including the pandemic which hinders the citizens from leaving their homes. However, there are several issues with current chatbots as they are unable to respond accurately to long queries. Furthermore, it has a significant response time which would discourage users from using chatbots. The aim of this study is to build a chatbot that efficiently handles extensive queries from users by providing contextually relevant responses. Long Short-Term Memory (LSTM) models are well-suited to handle long-term dependencies while Gated Recurrent Units (GRU) models are more efficient hence a hybrid model of GRU-LSTM is proposed as a solution. The performance evaluation metrics used are Bilingual Evaluation Understudy, BLEU and response time. The LSTM model obtains the highest BLEU score while the GRU model has the shortest response time. The proposed model has the second-best BLEU score outperforming the GRU model and the second-best response time outperforming the LSTM model. Hence, the proposed model is a good compromise between the two models as it has a reasonable BLEU score accuracy and response time.

Keywords: chatbot, Long Short-Term Memory, Gated Recurrent Units, Hybrid, Deep Learning

1. Introduction

Suruhanjaya Syarikat Malaysia (SSM) is a Malaysian statutory agency that supervises companies and businesses. As of today, there are only three ways to contact and clear a query with SSM which include directly going to the office, calling their hotline or emailing the respective person in charge. By September 4, 2021, Malaysia was ranked third in Southeast Asia for the number of cases and deaths due to COVID-19 with over 1,800,000 confirmed COVID-19 cases, over 250,000 active cases, and above 17,800 deaths (Rampal et al., 2020). This situation has raised many concerns about the safety of citizens when traveling out of their homes to public places such as Suruhanjaya Syarikat Malaysia (SSM)'s office. The other methods are time-consuming and tedious work for one to do for a simple query. The utilization of service-based chatbots would be efficient and prevent people from leaving their homes during the pandemic.

A chatbot is a computer program that is implemented to communicate with clients via natural language (Karri and Kumar, 2020). Recent AI advancements and the myriad amount of available data has allowed chatbots to perform more complex tasks. Chatbots can replace humans for monotonous jobs such as answering queries and giving efficient responses. Government agencies have begun adopting these technologies as a result of the positive impact of chatbots in the private sector. The chatbots are utilized to take on significantly complex tasks in diverse domains, e.g. health, social, welfare, public safety, taxation, and education. Chatbots eliminate the need for customers to wait for long periods to solve their queries and the requirement to travel to the SSM office. It helps to reduce the need to have human-to-human contact to clarify their queries, which will be helpful in the case of Covid-19 pandemic. It can also be beneficial even if the pandemic is curbed as chatbots can save people's time.

Although chatbots are potentially very beneficial to their users, current chatbots are unable to respond to long queries accurately. In a report surveyed above 700 people around the country in April-May 2021, more than half the respondents stated that the chatbots could not comprehend the queries while 45% of them felt frustrated because the chatbots gave irrelevant responses to the queries (Tewari, 2021). One of the possible reasons for this is the length of the input from users was too long and complicated. Thus, a service-based chatbot needs to have dialogue abilities to keep the customers engaged and respond to long queries. A generative chatbot is the solution as it can respond to long queries as well as converse with the user.

In general, chatbots have bad response time which leads to unhappy and disappointed customers. In addition, 47% said chatbots did not generate accurate answers and took an extended time to process and reply to the query (Tewari, 2021). This is an issue that needs to be addressed; however, present research papers compare models for their training time and accuracy but not much attention has been given to their response time. For these reasons, the paper aims to address the issue of handling long queries and response time of the chatbots.

The structure of the paper is as follows: the first section discusses the concept of a chatbot and explains the study objective. Next, the background study in the area of chatbots relevant to long-term dependencies and response time done by other researchers are explained in Section 2. Followed by the explanation of the deep learning models are explained in Section 3. The proposed approach and each component are described in Section 4. Subsequently, Section 5 discusses the result and its analysis. Finally, section 6 provides a conclusion for the conducted experiment and proposed future works that can be performed to better the research further.

2. Related Work

This section discusses the work done by previous researches based on two aspects of the chatbot. Section 2.1 explains current approaches to handle long queries while Section 2.2 is on response time. Section 2.3 describes the research gap that exists.

2.1 Approaches to Handle Long Queries

(Patil et al., 2020) proposed a crossbreed LSTM-based Ensemble model to handle long-term dependencies and retain the information in specific situations. LSTM is particularly suited to tackle long-term dependency difficulties found in chatbots since it has explicitly expanded memory capability (Patil et al., 2020). The Cornell Movie Dialogue corpus was used to train the model. The concept is to build several LSTM networks with differences in hyperparameters as part of the ensemble model. The member models run at the same time, and their individual outputs are aggregated to produce the overall model's output. The finding was that neither Ensemble LSTM nor Ensemble GRU performed significantly better than each other. For future study, further improvements can be achieved in terms of word embedding that is not limited by the knowledge base, developing a flexible and precise conversational model, and successfully simulating human conversation without the need for human interaction.

(Sojasingarayar A, 2020) suggested using Deep Neural Network (DNN) and Recurrent Neural Network (RNN) with Deep Reinforcement Learning (DRL) as a concept to be used for developing long conversation chatbots capable of creating an emotional bond with the user. The dataset used is the Cornell Movie Dialog Corpus. (Sojasingarayar A, 2020) used an encoder-decoder attention mechanism design to create a Seq2Seq AI Chatbot. This encoder-decoder employs an LSTM-based Recurrent Neural Network. The model's performance was restricted in extended conversations, the output was repetitive and generic, and the chatbot performed below optimal for replicating human interaction due to a lack of real-life quality data. Furthermore, many sentences were eliminated owing to their length or inconsistency.

2.2 Approaches to Improve Response Time

(Janati et al., 2020 Miklosik et al., 2021 and Kuhail et al., 2022) suggested a framework that utilizes Natural Language Processing (NLP) and Keywords Vote++ (keyword extraction method) as a chatbot backbone for e-learning to reduce the chatbot's response time. The process of keyword extraction entails finding the words and phrases that reflect the document's major topics. The researchers concluded that selecting terms based on keywords allows for the establishment of a more homogenous tree structure for indexing multimedia material while also reducing execution time. The limitations include its restricted capability, which prevents it from successfully interacting with the learner when they ask generic inquiries. In order to broaden the range of chatbot interactions, future development will work on adding a Chit-Chat to replicate a human discussion and implementing speech recognition on the chatbot.

(Chen et al., 2022) proposed a hybrid architecture that combines recurrent neural networks with Bidirectional Encoder Representations from Transformers (BERT) to reduce computation time while maintaining accuracy. Transformer-based model and the RNN-based model can't be integrated directly. Hence, initialize the decoder hidden state for the RNN-based sequence to sequence model (Chen et al., 2022). To eliminate the additional work, an easier way of computing the average of BERT's output was utilized. For training, the model is fed the NLPCC 2018 grammatical error correction (GEC) dataset. Bilingual Assessment Understudy Score (BLEU), inference speed, and training speed are the three evaluation measures. In all the trials, BERT-GRU had the highest BLEU Score. (Castro et al., 2022) also used BLEU score to measure the efficiency of the encoder-decoder architecture.

2.3 Research Gap

LSTM model cannot be utilized to improve chatbot response time. The LSTM model requires high computational power and is not very efficient though it is proven to be one of the most suitable models to build a chatbot. This model can handle long-term dependencies but due to its complex nature, it cannot help to reduce the response time of the chatbot. On the other hand, GRU models are not as powerful as LSTM models and only perform well when the dataset is small and consists of long text. Although GRU models are more efficient than LSTM models and thus can be used to improve the chatbot’s response time, they are inferior to the latter for many scenarios including handling long-term dependencies.

Other than that, the hybrid model concept has never been implemented in the chatbot domain although it has been proven successful in other applications. It has been used to improve inference time in a Chinese Sentence Correction algorithm and it is as powerful as independent models. Hence, this study plans to build a hybrid model of GRU-LSTM for the chatbot in order to combat the long queries and response time problem.

3. Deep Learning Models

This section discusses the proposed model theoretically. We begin by explaining the framework of the standard LSTM and GRU models and proceed with the proposed model.

3.1 LSTM

Long Short-Term Memory (LSTM) networks are Recurrent Neural Networks (RNN) modifications with intentionally enhanced memory capability that is ideally equipped to manage long-term dependencies (Patil et al., 2020). The LSTM integrates context information in a gated cell. To regulate the data to be written, stored, read, and erased, the cells utilize Forget, Input, and Output gates, which are deployed by sigmoid using element-wise multiplications (Patil et al., 2020). The forget gate learns the weights that affect the decay rate of values stored in memory cells. While the input and output gates are off and the forget gate is not generating decay, the memory cell maintains its value over time, causing the error gradient to remain constant during backpropagation. As a result, the model can recall data for longer periods of time. In the field of conversational agents for time series discussions, LSTM has been proven to work well and preserve the context for longer periods of time. Figure 1 illustrates the LSTM's overall design. LSTM equations are shown in Figure 2.

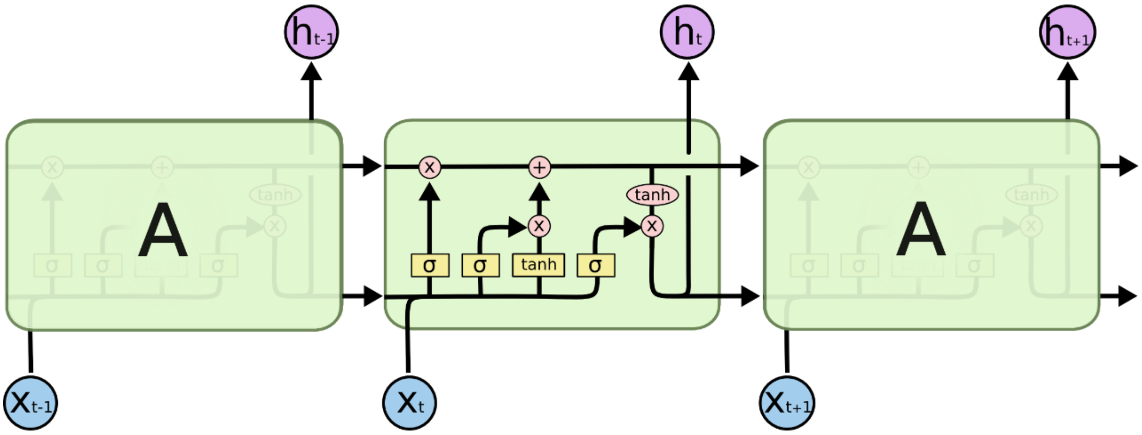
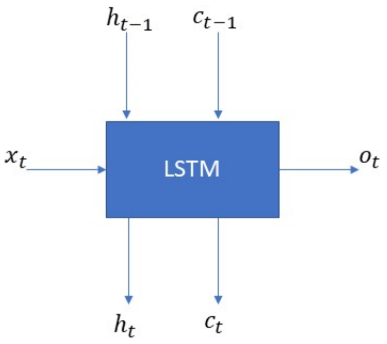


Figure 1. LSTM model structure (Olah, 2015)



$$f_t = \sigma_g\ (W_f \times\ x_t + U_f \times h_{t-1} + b_f)$$
$$i_t = \sigma_g\ (W_i \times\ x_t + U_i \times h_{t-1} + b_i)$$
$$o_t = \sigma_g\ (W_o \times\ x_t + U_o \times h_{t-1} + b_o)$$
$$c'_t = \sigma_c\ (W_c \times\ x_t + U_c \times h_{t-1} + b_c)$$
$$c_t = f_t \cdot c_{t-1} + i_t \cdot c'_t$$
$$h_t = o_t \cdot \sigma_c(c_t)$$

σ_g : $sigmoid$
σ_c : tanh
. : element wise multiplication

f_t $is\ the\ forget\ gate$
i_t $is\ the\ input\ gate$
o_t $is\ the\ output\ gate$
c_t $is\ the\ cell\ state$
h_t $is\ the\ hidden\ state$

Figure 2 LSTM equations for a single timestep (Rastogi, 2021)

Where f_t is represents Forget Gate, it, and o_t represents the input gate and output gate, cell state and hidden state are represented as c_t and h_t. Sigmoid and tanh are represented as σ_g and σ_c. The new input and weights are indicated as x_t and W/U.

3.2 GRU

A gated recurrent unit (GRU) is a Recurrent Neural Network (RNN) model. It enables the recurrent unit to record associations over several time steps (Patil et al., 2020). Unlike LSTMs, GRU contains two gates to govern the flow of information and improve the outputs: Reset and Update. When compared to LSTM, the update gate may be thought of as a mixture of the Forget and Input gates. The update gate specifies how much data from previous time steps should be sent on to the next state. This provides GRU an advantage over LSTM since it may choose to keep all features from previous timestamps. The reset gate is used to identify the information that should be deleted because it is irrelevant. GRU (Lendave, 2021). It utilizes fewer training parameters, uses less memory, and runs quicker than LSTM. Figure 3 depicts GRU's general architecture. GRU model's equations are illustrated in Figure 4.

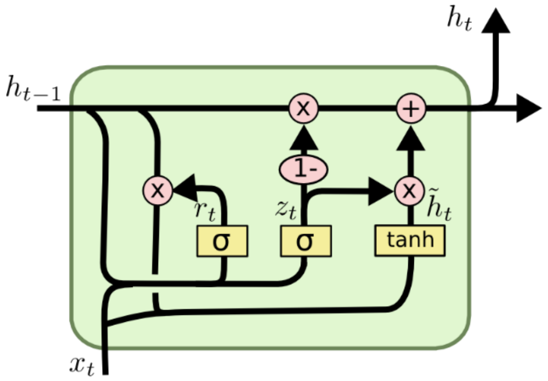


Figure 2. GRU model structure (Mani, 2019)

$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Figure 3. GRU equations for a single timestep (Mani, 2019)

Where Z_t and r_t stands for update and reset gates. The hidden state and new input are represented as h_t and x_t.

4. Proposed Approach

This section explains the methods and methodology utilized to implement the proposed model in this study. It begins with data gathering followed by data pre-processing, and model building. The overall flow can be seen in Figure 5.

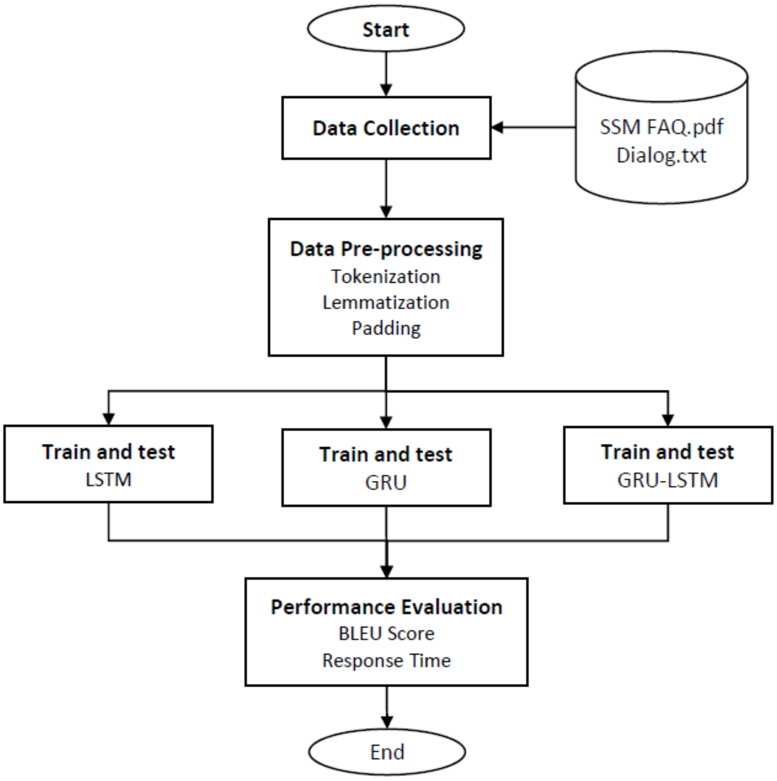


Figure 4 Overall flow of implementation of model

4.1 Data Gathering

For this study, two different datasets will be used to train our models. One text file from Kaggle's website titled 'dialog.txt' and the other is a list of questions and answers from Suruhanjaya Syarikat Malaysia's FAQ on their official website (SSM,2022). The data from the FAQ pdf file was scrapped using the tabula library and further refined using Bytescout PDF Multitool.

4.2 Data Pre-processing

Both datasets are loaded into the Python IDE for pre-processing. The first step is to remove all non-alphanumeric terms such as emojis, symbols and special characters. This is done using simple for loops and regex functions. The flowchart is shown in Figure 6. Once the data is free of special characters, the questions and answers are zipped into pairs.

Then another regex function was used to remove punctuations such as '!', '?', and '.'. Each lines of the pairs are then added into separate arrays to split input and target sentences. Input sentences are the query for the chatbot while target sentences are the chatbot replies. Then we add the terms <START> and <END> to each sentence. The two characters are added at the front and end of the target sequence to help the model understand where to begin and end the text generation. Splitting up each sentence into words and adding each unique word to the vocabulary set is done. This algorithm was applied to both input and target sentences.

Next, a dictionary was created to store the input words and target words as key-value pairs, with the word as the key and the index as the value. This step is necessary in order to help the model understand the words since computers only understand the numbers. To decode the sentences, the reverse features dictionary was created in which the word and index were stored in reverse compared to the initial dictionaries. Three arrays were created using the data obtained. Then a nested for-loop is used to convert the value of '0' to '1' whenever the word is present in that respective line. It utilizes the dictionaries created to identify the value of the matrix that needs to be converted.

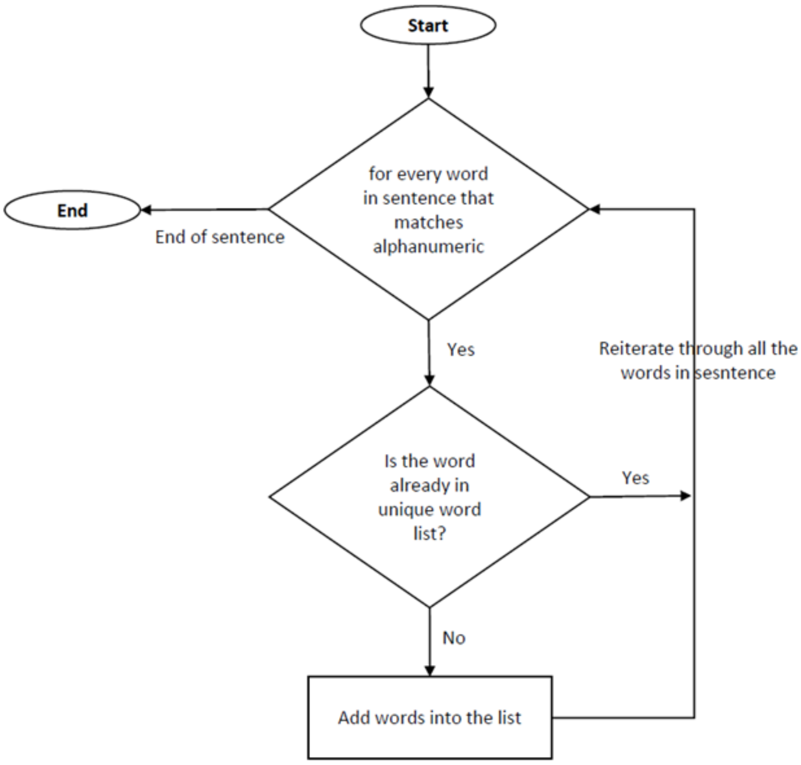

Start
for every word in sentence that matches alphanumeric
End
End of sentence
Yes
Reiterate through all the words in sesntence
Is the word already in unique word list?
Yes
No
Add words into the list

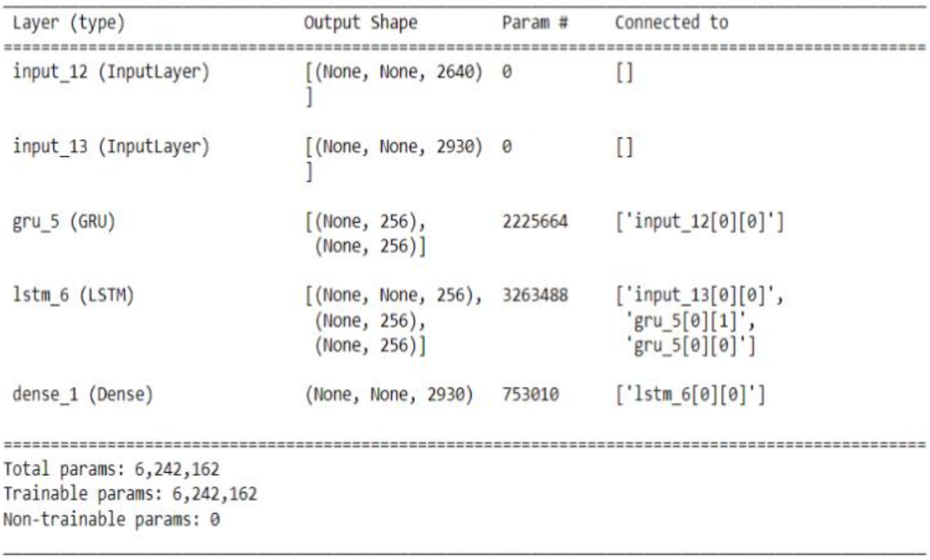
Figure 6. Flowchart of alphanumeric removal algorithm

4.3 Building Hybrid Model

We proposed combining GRU and LSTM algorithms as a hybrid model for our text generation chatbot, which is based on the seq2seq approach. The approach involves two models, one known as the encoder to encode the input sequence, and a second, a decoder to decode the encoded input sequence into the target sequence. The models were imported from the 'Keras.layers' library. GRU was used as the encoder while LSTM was the decoder of this hybrid model. The encoder model requires an input layer that defines a matrix for holding the one-hot vectors and a GRU layer with some number of hidden states. The decoder model structure is much like the encoder's structure. The decoder model's initial state is given the encoder's hidden state and outputs along with decoder inputs. The number of inputs and outputs by the encoder GRU and decoder LSTM differentiates the encoder-decoder approach from the standard model. Figure 7 shows the model summary. The 'optimizer' used for the model is 'rmsprop' while the loss function chosen was 'categorical_crossentropy' and 'sample_weight_mode' used was 'temporal'. The performance of the model is measured using BLEU score (Castro et al., 2022 and Doshi 2021).

5. Results

In previous section, the research methodology and implementation were discussed in detailed to carry out this research. In this section, the result obtained from the implementation will be discussed and justified.



```
___________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
===================================================================================================
input_12 (InputLayer)           [(None, None, 2640)  0           []
                                ]

input_13 (InputLayer)           [(None, None, 2930)  0           []
                                ]

gru_5 (GRU)                     [(None, 256),        2225664     ['input_12[0][0]']
                                 (None, 256)]

lstm_6 (LSTM)                   [(None, None, 256),  3263488     ['input_13[0][0]',
                                 (None, 256),                     'gru_5[0][1]',
                                 (None, 256)]                     'gru_5[0][0]']

dense_1 (Dense)                 (None, None, 2930)   753010      ['lstm_6[0][0]']

===================================================================================================
Total params: 6,242,162
Trainable params: 6,242,162
Non-trainable params: 0
___________________________________________________________________________________________________
```

Figure 5. Hybrid model summary

5.1 Performance of GRU Model

The standard GRU model built with encoder GRU and decoder GRU has a dimensionality of 256 and was trained with 300 epochs. The model got a BLEU score of 0.5070. Hence, the model can get about 50 % of the generated reply correct. As for the response time, the GRU model took 0.4546 milliseconds to respond on average when calculated with all given inputs. The details of the model and the results can be seen in Table 1.

5.2 Performance of LSTM Model

The standard LSTM model built with encoder LSTM and decoder LSTM has a dimensionality of 256 and was trained with 300 epochs. The model got a BLEU score of 0.5896. Hence, the model can get about 58% of the generated reply correct. As for the response time, the LSTM model took 0.5725 milliseconds to respond on average when calculated with all given inputs. The details of the model and the results can be seen in Table 1.

5.3 Performance of Hybrid Model

The proposed hybrid model built with encoder GRU and decoder LSTM has a dimensionality of 256 and was trained with 300 epochs. The model got a BLEU score of 0.5597. Hence, the model can get about 55% of the generated reply correct. As for the response time, the hybrid model took 0.5697 milliseconds to respond on average when calculated with all given inputs. The details of the model and the results can be seen in Table 1.

Table 1 Performance evaluation of the models

Model	BLUE Score	Response Time
GRU	0.5070	0.4546
LSTM	0.5896	0.5725
Hybrid	0.5597	0.5697

5.4 Overall Performance

The LSTM model got the highest BLEU score. However, the proposed hybrid model performed better compared to the standard GRU, which got a lower BLEU score when compared with LSTM. The comparison can be seen in Figure 8 in which the LSTM model outperforms both the hybrid and GRU models. The BLEU score indicates that the LSTM model can replicate the answers from the dataset the best. As for response time, the GRU model takes the least amount of time compared to the other two models with the LSTM model taking the longest. It shows the efficiency of the GRU model which is illustrated in Figure 9. Table 2 shows the overall performance of the models.

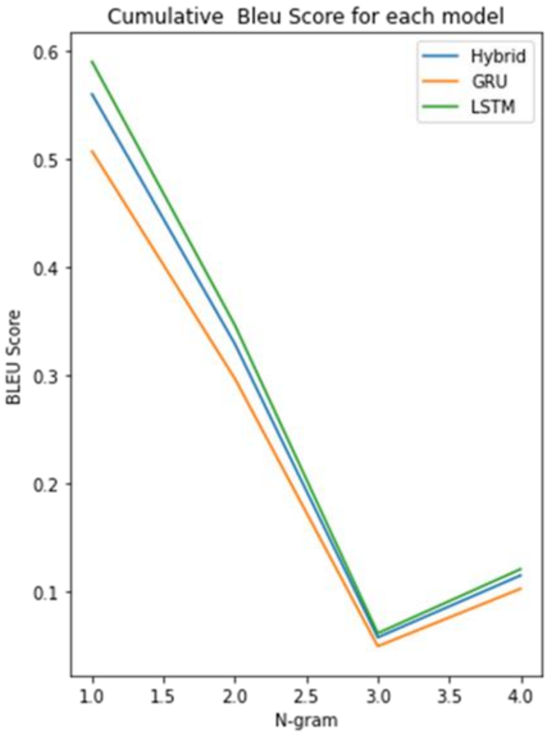


Figure 8. Cumulative BLEU score for models

Table 2. Overall performance summary

Evaluation metrics		Models		
		GRU	LSTM	Hybrid
Cumulative BLUE Score (/1)	1-gram	0.5070	0.5896	0.5597
	2-gram	0.2969	0.3463	0.3290
	3-gram	0.0498	0.0618	0.0579
	4-gram	0.1029	0.1210	0.1152
Response Time (ms)		0.4546	0.5725	0.5697

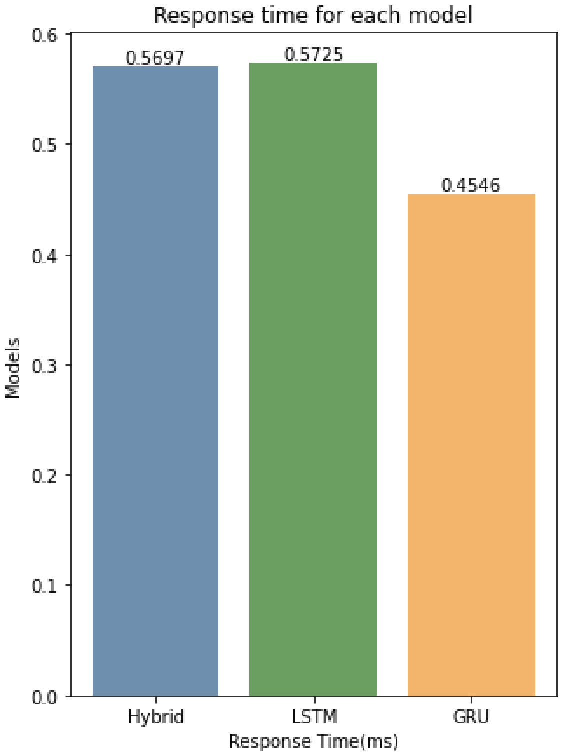


Figure 9. Response time for models

6. Conclusion

In this research, a noble model of GRU-LSTM was introduced as a chatbot model for Suruhanjaya Syarikat Malaysia. Despite training the model with SSM FAQ dataset and a dialog dataset, the model has not performed well. The BLEU score was better than the GRU model but LSTM slightly outperforms it. As for the response time, it was marginally faster than the LSTM model but GRU model had clearly the shortest response time. This research has proven that the hybrid model has relatively good accuracy and a comparably decent response time. The standard LSTM is the best in terms of accuracy but the GRU is the best in terms of efficiency. Future work may be done by experimenting with the hyperparameters. Since all the models scored a low score, the issue could be the low number of data fed into these models and the lack of an attention layer like Luong Attention or Bahdanau attention. Hence, these additional details could be looked into further.

Acknowledgements

We would like to thank Multimedia University for supporting the project with the required funds (Ref: MMUI/220096).

References

Castro, R., Pineda, I., Lim, W., & Morocho-Cayamcela, M. E. (2022). Deep learning approaches based on transformer architectures for image captioning tasks. *IEEE Access*, *10*, 33679-33694.

Chen, J. W., Sigalingging, X. K., Leu, J. S., & Takada, J. I. (2020). Applying a hybrid sequential model to Chinese sentence correction. *Symmetry*, *12*(12), 1939.

Doshi, K., "Foundations of NLP Explained — Bleu Score and WER Metrics," Towards Data Science, 11 May 2021. [Online]. Available: https://towardsdatascience.com/foundations-of-nlpexplained-bleu-score-and-wer-metrics-1a5ba06d812b.

El Janati, S., Maach, A., & El Ghanami, D. (2020). Adaptive e-learning AI-powered chatbot based on multimedia indexing. *International Journal of Advanced Computer Science and Applications*, *11*(12).

Karri, S. P. R., & Kumar, B. S. (2020, January). Deep learning techniques for implementation of chatbots. In *2020 International Conference on Computer Communication and Informatics (ICCCI)* (pp. 1-5). IEEE.

Kuhail, M. A., Thomas, J., Alramlawi, S., Shah, S. J. H., & Thornquist, E. (2022, October). Interacting with a Chatbot-Based Advising System: Understanding the Effect of Chatbot Personality and User Gender on Behavior. In *Informatics* (Vol. 9, No. 4, p. 81). MDPI.

Lendave, V., "LSTM Vs GRU in Recurrent Neural Network: A Comparative Study," Developers Corner, 27 August 2021. [Online]. Available: https://analyticsindiamag.com/lstm-vs-gru-inrecurrent-neural-network-a-comparative-study/.

Mani, K., "GRU's and LSTM's," Towards Data Science, 18 February 2019. [Online]. Available:https://towardsdatascience.com/grus-and-lstm-s-741709a9b9b1.

Miklosik, A., Evans, N., & Qureshi, A. M. A. (2021). The use of chatbots in digital business transformation: a systematic literature review. *IEEE Access*, *9*, 106530-106539.

Olah, C., "Understanding LSTM Networks," colah's blog, 27 August 2015. [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

Patil, S., Mudaliar, V. M., Kamat, P., & Gite, S. (2020). LSTM based Ensemble Network to enhance the learning of long-term dependencies in chatbot. *International Journal for Simulation and Multidisciplinary Design Optimization*, *11*, 25.

Rampal, L., Liew, B. S., Choolani, M., Ganasegeran, K., Pramanick, A., Vallibhakara, S. A., ... & Hoe, V. C. (2020). Battling COVID-19 pandemic waves in six South-East Asian countries: a real-time consensus review. *Med J Malaysia*, *75*(6), 613-625.

Rastogi, M., "Tutorial on LSTMs: A Computational Perspective," Towards Data Science, 27 January 2021. [Online]. Available:https://towardsdatascience.com/tutorial-on-lstm-acomputational-perspective-f3417442c2cd#0d00.

Sojasingarayar, A. (2020). Seq2seq ai chatbot with attention mechanism. *arXiv preprint arXiv:2006.02767*.

SSM (2022) "FAQ," Online]. Available: https://www.ssm.com.my/Pages/FAQ/FAQAll. aspx.

Tewari, S., "Report: Consumers face challenges using chatbots to resolve queries," The Star, 24 June 2021. [Online]. Available:https://www.thestar.com.my/tech/technews/2021/06/24/report-consumers-facechallenges-using-chatbots-to-resolve-queries.