BilinBot: A Bilingual Chatbot using Deep Learning

Sanjeda Sara Jennifer
Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
jennifer.sanjedasara@gmail.com

Sheikh Aminul Islam
Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
2015-3-65-013@std.ewubd.edu

Sharmin Sultana Koly
Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
2015-3-65-010@std.ewubd.edu

Rashedul Amin Tuhin
Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
mcctuhin@ewubd.edu

M. Saddam Hossain Khan
Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
saddam.cse@ewubd.edu

Md. Mohsin Uddin
Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
mmuddin@ewubd.edu

2023 5th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA) | 979-8-3503-3752-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/HORA58378.2023.10156681

***Abstract*— Chatbots are widely famous all over the world. Businesses of all scales, web services, customer supports and various others are integrating chatbots. However, most of them work in the English language. Our research aims to create a bilingual chatbot that can be used by people from different regions and converse in both Bangla and English. It will not just enable seamless communication in both languages, but enhance user experience for people of all fields and regions in Bangladesh. We named our chatbot "BilinBot" which comes from the idea of creating a "Bilingual Chatbot". Our BilinBot can take questions as user input in English and Bangla language, predict the answer and respond accordingly as it has been trained. Natural language processing approaches are used to process the data. We performed ROUGE-1, ROUGE-2 and ROUGE-L evaluations to assess BilinBot. Our experimentation indicated that Google BERT is the best feature engineering approach for processing natural language, and for the training phase, LSTM and GRU performed best. We compared BilinBot with the state-of-art works and achieved better performances.**

Keywords*— *Chatbot, Natural Language Processing, Bilingual Chatbot, Artificial intelligence, Machine learning, Deep learning, Transfer learning

I. INTRODUCTION

Artificial Intelligence is the usage of technology to mimic human activity or behaviour. Alan Turing suggested the idea of computers conversing with humans and introduced the Turing test [1]. This began the study of conversational agents, a method of interaction that not only performs the processing of natural language but also responds using human language. Eliza [2] was the first intelligent conversational agent which was created to demonstrate the communication between machines and humans at the Massachusetts Institute of Technology (MIT).

A chatbot is an artificial intelligence software agent capable of creating a conversation between a virtual agent and a human [3], whether it is text-based, spoken, or non-flask communication [4]. However, maximum chatbots and research regarding this is based on mostly the English language due to the lack of proper tools and appropriate resources. We propose BilinBot, an intelligent bilingual chatbot that allows conversation in both Bangla and English. Bangla is widely spoken as it is the mother tongue of the people of Bangladesh and also the language of some parts of India such as West Bengal. Almost 228 million people speak Bangla [5], as it is the fourth language in the world, so most of the organizations in our country use both languages. There is a lack of appropriate resources and a lot of work has to be done as it is considered a low-density language. Considering the situations and demands in the present day, we have been motivated to develop BilinBot.

A multilingual chatbot offers support for users in multiple languages during a live chat [6]. It provides efficient solutions in the chat initiator's preferred language. Many industries, entrepreneurs and business organizations are trying to be more productive using chatbots as they are accessible anytime, cost-effective, and work automated. Nowadays chatbots have been used in several online services like online shopping conversations, ordering food online, and customer services like hospital appointments, FAQs, etc.
The main challenge in chatbot research is finding an appropriate dataset. Making a perfect chatbot is quite difficult as it needs a large dataset and should be cleaned according to the model. Collecting datasets for the English language is not that complicated as it is the first language in most countries, but finding a good dataset in other languages like Bangla is very difficult. Our work aims to design a bilingual chatbot using the best deep learning and natural language processing approaches with the highest accuracy score. Therefore, the primary objectives of this research are:

- Analyze the finest feature engineering approaches
- Identify the best deep-learning method
- Prepare grammatical and spelling error-free datasets
- Construct a bilingual chatbot model

The rest of our paper is organized as follows: Section II represents the related work, Section III gives the materials and methods used for our BilinBot followed by Section IV, which presents the result analysis. Finally, Section V concludes the paper by presenting the limitations of our work and the future scope of our project.

II. RELATED WORK

Chatbot, a conversational agent can converse with humans based on the provided data and trained knowledge using natural language processing (NLP). There have been many studies and research related to the rising demand for chatbots. For example, Amiri et al. [7] examined how chatbots were used in the public health response to the Covid-19 outbreak.

979-8-3503-3752-5/23/$31.00 ©2023 IEEE

They identified 61 chatbots from 30 countries, six types of public health response and 15 use cases about the utilization of chatbots. Dahiya et al. [3] highlighted the design and deployment of a chatbot architecture to explore its applications in various fields.

Nowadays, along with the NLP advances, BERT is a multilingual model that can manage a hundred languages. In reference [8], the paper demonstrates that multilingual BERT can solve complex Question Answering tasks described in the English Question Answering SQuAD dataset which can be accomplished in Japanese and French.

In reference [9] the paper shows the design of a chatbot which can be used for mental health counseling. They made a demo chatbot using interactive emojis and GIFs so that the user experience can be improved while searching for online self-help tips.

Chatterbot, a conversational agent, has embedded data to decide the sentences and construct a decision itself, as an answer to a query. In reference [10] the target was to make a Human-to-Machine conversation modeling which will be a knowledge-based model in a dataset.

Single responses such as taking sequences of words and sentences have been done widely. In reference [11] they made a human-computer conversation that will respond as a multi-turn.

In reference [8], the authors explored retrieval-based chatbots in the issue of response choice for long conversations. The matched vector was obtained using RNN.

Many chatbots were made using pattern-matching algorithms. The first chatbot Eliza [2] and reference [10] show chatbot using pattern-matching techniques. These types of chatbots are mainly knowledge-based chatbots. Usually, they take sentences as input from the user and then calculate similarity.

In reference [8] a conversational agent was built using BERT and they used RNN for training. Similarly in reference [12], the authors used RNNs to build their chatbot. But they implemented an encoder-decoder attention mechanism which is unlike our method. We developed a chatbot using deep neural networks and utilized RNN, GRU, LSTM to analyze and find the best approach.

Ranoliya et al. [13] design and development a chatbot to answer the frequently asked questions in for universities using Artificial Intelligence Markup Language (AIML) and Latent Semantic Analysis (LSA). The interactive chatbot is designed to provide information about college admission, the university environment, academic information, and more. The paper also provides an overview of existing chatbot applications and AIML tags.

Nallamani et al. [14] developed a chatbot that could be easily tailored to suit diverse circumstances and deployed in various sectors as long as relevant data is available. It was built using natural language processing and deep learning techniques through Python and PyTorch. The article also discusses the dataset used to train the model, the neural network architecture, and the hyperparameters used for training.

Solanki et al. [15] discussed the development of smart chatbots and their use in healthcare. They aimed to design a chatbot that provides a more convenient and accessible method of delivering healthcare information and services to patients. They did it by using various components and algorithms in the design of the chatbot, including conversational artificial intelligence and natural language processing technology.

III. Background

The development of conversational agent systems has been growing. Rule-based, traditional techniques were deployed earlier to create chatbots, but now with the advancement of technology, Natural Language Processing (NLP) [16] and deep learning methods, such as, deep neural networks [17] are being used to train a chatbot.

Google BERT (Bidirectional Representation for Transformers) [18] is a natural language processing technique. The greatest addition in BERT is the ability of bidirectional training. In the process to train the BERT, the model was given two sentences, and it had to learn to predict whether or not the original document the second sentence is subsequent of the first sentence. This process is called Next Sentence Prediction (NSP) [19]. A sequence of words is given in language modeling and it was expected to predict the next word from the sequence. A traditional language model predicts every next token but in Masked Language Modeling (MLM) a percentage of randomly picked input tokens are masked and the model only predicts those tokens [18].

Recurrent Neural Network (RNN), a variant of the neural network family has persistence in learning [20], [21]. RNN loop through the previous steps and tries to relate and understand. The looping allows information to persist. Fig. 1 shows a loop in a segment of a recurrent neural network where the input is represented by "xt" and the current state is represented by "ht". The loop 'A' allows transferring information from one step to the next step of the network. This just does not happen once.

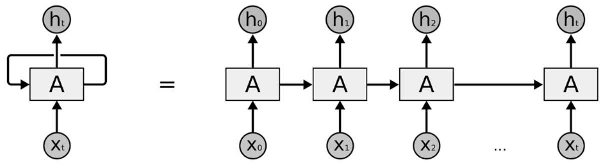

Fig. 1: The sequence of looping in a recurrent neural network.

Fig. 1 shows the sequence of transferring information in a recurrent neural network, where ht, xt, ht-1 are the current state, the current input and previous state respectively. In every state, a recurrent neural network relates information with the information of the previous step [23].

LSTM (Long Short-Term Memory) [24] is an architecture of RNN, which can learn long-term dependencies and remembers information for a long period. The vanishing gradient problem of RNN is solved in the LSTM network as it has 3 gates: input gate, output gate, and forget gate [25]. The current inputs are handled by the input gate and irrelevant information is discarded by the output gate. The output gate produces predictions.

Fig. 2 shows the input gate. It finds which input value to be used to write the memory. The Sigmoid function decides which values are to be transferred through (0 or 1). The tanh function gives weight to the transferred values from -1 to 1.

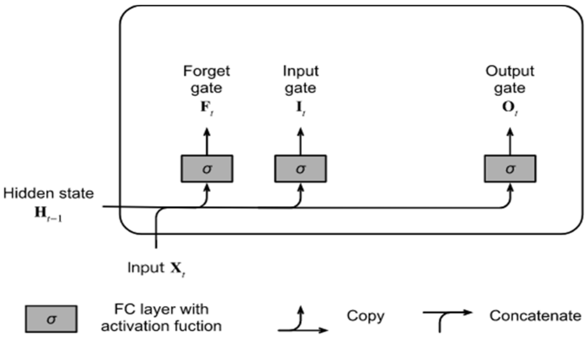

Fig. 2. The input, output, and forget the gate in an LSTM.

The output is determined by the input and the memory of the block. The forget gate determines which information to forget from the block, and is determined by the sigmoid function.

Gated Recurrent Unit (GRU) [26] is a gating mechanism in recurrent neural networks. GRU is an improved version of recurrent neural networks. GRU solves the vanishing gradient problem of recurrent neural networks [27]. It has an update gate and reset gate, which are separated in a fully gated unit of GRU. Fig. 3 shows the mechanism of a fully gated unit where xt, ht and zt are the input, output and reset gate vector respectively. The reset gate decides how much information should be forgotten and the update gate determines the information that should be pushed forward [28].

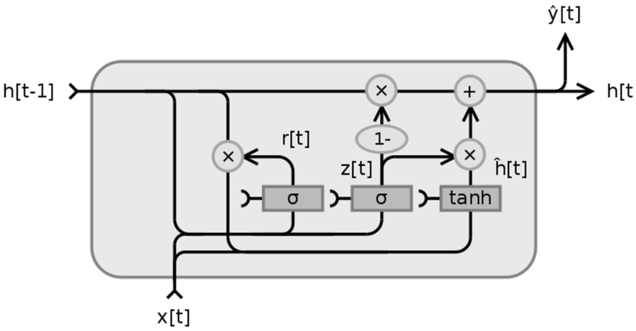

Fig. 3. The mechanism of GRU.

IV. MATERIALS AND METHODS

In this research, we are proposing a deep learning-based bilingual chatbot that is easily maintainable and can interact with users in English and Bangla language. The chatbot takes questions as user input, then it predicts an answer using a trained model and responds accordingly. Fig. 4 shows the working procedure of the chatbot.

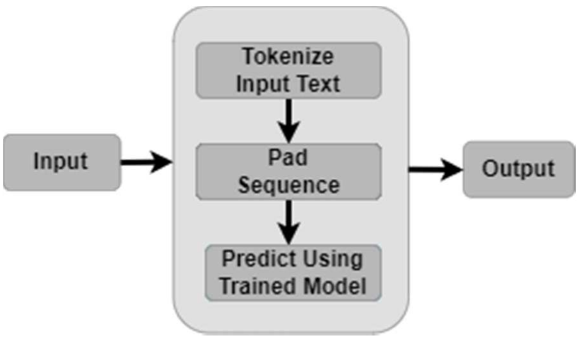

Fig. 4. Working Procedure of the Chatbot

A. Dataset

Our English datasets contain conversation data, questions and answers about different topics and categories such as computers, emotion, food, greetings, health, history, humor, literature, movies, politics, psychology, science, sports, etc.

We paraphrased every question of the dataset to find out every possible form of the questions so that the model can understand all the ways a question can be asked. After paraphrasing, many duplicate data were generated, which we removed from the dataset. For the entire dataset, we checked for grammartical and spelling errors and fixed them. We generated the Bangla dataset by translating the English dataset.

B. Data Preprocessing

The data for the chatbot is textual data which cannot be used for training in the neural networks directly. Hence, we need to process the textual data and generate numeric data for the training. We used the Tokenizer API of Keras for tokenizing. The Tokenizer API vectorizes the text corpus and creates a vocabulary. Then we can use the Tokenizer to convert the text data to a sequence of integers where each integer represents the index of words in the vocabulary. To tokenize using the Tokenizer, first, we fitted the tokenizer on the questions. Then using the fitted tokenizer, we converted all the questions from texts to sequences. The length of the questions was different so the sequences were padded to make the length of the sequences same. After padding the numeric sequences, the data was ready for training the model.

The "Term Frequency Inverse Document Frequency" (TF-IDF) Vectorizer transforms text into numeric data. For all the words in a corpus, the TF-IDF Vectorizer computes the TF-IDF score. It computes the term frequency and inverse document frequency and calculates the TF-IDF score and vectorizes the data. To implement TF-IDF Vectorizer we used the API of Scikit-learn. and took the N-gram range 1 to 3. That means, it generated unigram, bigram, and trigram features. We fitted the vectorizer for all the questions of the dataset and then exported the train data from the vectorizer.

To implement word2vec [29], we pre-processed the data, trained the word2vec model, generated an embedding matrix using that word2vec model, and used the embedding matrix in the embedding layer of the neural network. Using the text to word sequence method of Keras, we converted each sentence of the dataset to word sequence. We took the embedding size 128 and trained the word2vec model using the data. For each word of the vocabulary, we generated an embedding matrix using the word2vec model. In a Keras Neural Network model, we used the generated embedding matrix in the embedding layer.

BERT (Bidirectional Representation for Transformers) is another natural language processing tool which has the ability of bidirectional training. We used a pre-trained model of BERT. To implement BERT, at first, we loaded the pre-trained BERT layer model and the vocabulary file. We exported the vocabulary from the BERT Layer and used that data to create the FullTokenizer. Using the FullTokenizer API of BERT we tokenized all the questions and converted all tokens to IDs for all the tokenized questions.

C. *Model Training*

The next step after preparing the training data is training the model using deep learning algorithms. To implement RNN, we preprocessed and tokenized the data using a tokenizer, converted the texts to sequences of integers and padded the sequences to make them all of the same length. Then we build the RNN and performed multiple training for the combination of the different activation functions and optimizers. At first, we added the Embedding layer which maps each input word to a multidimensional vector and added the RNN layer. We added the Dense output layer where we added the activation function [30], and fitted the model for the training.

To implement LSTM, first, we tokenized the data using a tokenizer, build the LSTM Network, trained multiple LSTM models combining different optimizers and activation functions. To build the deep learning model, at first, we added the Embedding layer, added the LSTM layer and finally added the Dense output layer with the activation function. For every word of the vocabulary using the activation function, it produces a probability. We set the optimizer and compiled the model. Finally, we fitted the model for our training data. We took 250 epochs and 0.3 validation splits.

V. Result Analysis and Discussion

For evaluating the prediction of BilinBot, the ROUGE-Score python module was used. For the ROUGE-1 evaluation, which includes Precision, Recall, and F-score, we predicted the answers to each question of the test dataset. Then we compared each predicted answer with the original answer using the ROUGE-1 scorer and we got the ROUGE-1 scores for all the predictions. We also performed ROUGE-2 and ROUGE-L evaluation. We used ROUGE-2 and ROUGE-L API of the ROUGE-Score python module, which were performed in the same way.

A. *Result Analysis*

We trained and tested BilinBot using Word2Vec, Tokenizer (Keras), N-Gram TFIDF, Google BERT, and combining POS Taggers in Word2Vec. We performed the analysis to find out the best feature engineering approach for the chatbot. Fig. 5 shows the analysis of the result.

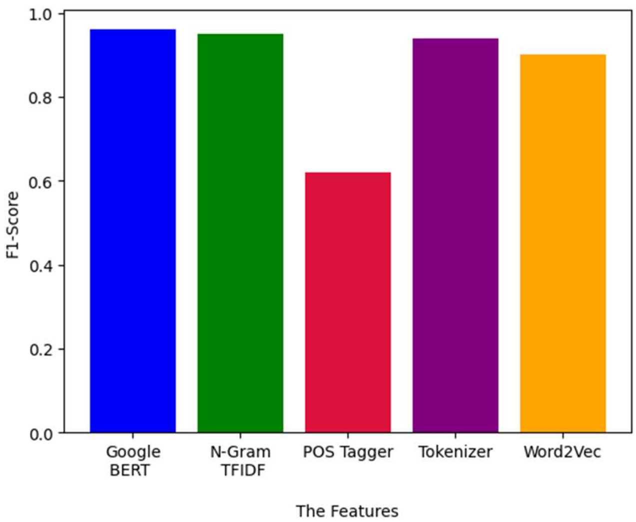

Fig. 5: F1 score of models for different feature engineering methods.

We got the highest F1 score of 0.96 (96%) for Google BERT, followed by 0.95 (95%) of TFIDF Vectorizer. For Tokenizer the F1 score was 0.94 (94%) and only 0.90 (90%) Word2Vec. The POS (Parts of Speech) Tagger had the lowest performance. From this analysis, we can say Google BERT is the best feature engineering approach for the chatbot.

We trained and tested the chatbot for different deep learning approaches as well. We used GRU, LSTM, RNN, and the traditional Neural Network. Their performances were compared and analyzed to find out the best model training approach for the chatbot. Fig. 6 shows the result of the analysis.

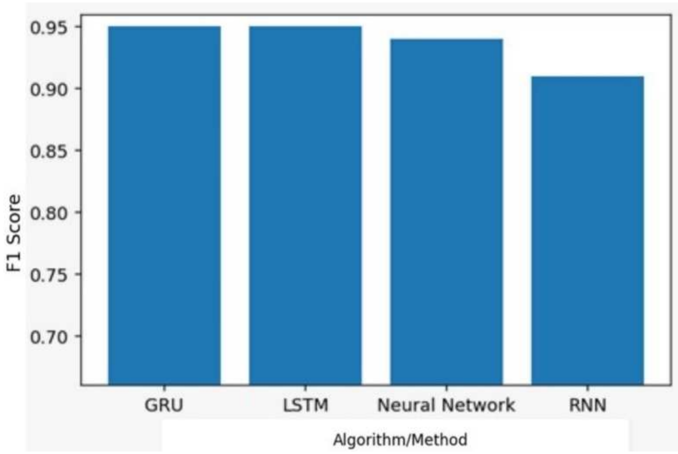

Fig. 6. F1 score of models for different feature engineering methods.

As the diagram depicts, we got better results for both LSTM and GRU. For both LSTM and GRU the F1 score was 0.95 (95%). For RNN the F1 score was 0.92 (92%).

On the trained models for the English dataset, we got a maximum 0.96 (96%) F1 score. The ROUGE scorer was used for the evaluation. During testing the model (trained using the English dataset) we performed ROUGE-1, ROUGE-2, and ROUGE-L evaluation. For ROUGE-1 and ROUGE-L, the F1 score was 0.96 (96%) and for ROUGE-2 the F1 score was 0.73 (73%).

On the trained models for the Bangla dataset, we got a maximum 0.76 (76%) F1 score. The ROUGE scorer was used for the evaluation. For ROUGE-1 and ROUGE-L the F1 score is 0.76 (76%) and for ROUGE-2 the F1 score is 0.73 (73%).

B. *Comparative Analysis*

As many works have been done before regarding chatbot, a comparison is needed to be performed to compare our work with previous existing state-of-art chatbots. We have chosen Conversational AI [12] and Seq2Seq AI [31], the two state-of-art work to perform the comparative analysis with our model.

The Conversational AI Chatbot was developed by implementing encoder-decoder attention mechanism architecture using two RNNs with LSTM cells. The encoder takes a sentence as input and at each timestep, it processes one symbol (word). The decoder is influenced by the thought vector and generates another sequence, one symbol at a time. They have introduced an attention mechanism called the Neural Machine Translation which lets the decoder to observe the input sequence while decoding.

The Seq2Seq AI also deployed the same attention mechanism architecture, encoder-decoder, using two RNNs with LSTM cells. They proposed to use CNN (Convolutional Neural Network) instead of RNN in the encoder. LSTM was used for the decoder enhancement technique.

Similar to Conversational AI [12] and Seq2Seq AI [31], our chatbot is also deep learning-based. Both of them used encoder-decoder mechanisms. But that mechanism was not used in our chatbot. For natural language processing, we used BERT, Word2Vec, and other feature engineering approaches. We proposed a solution that enabled our chatbot to understand the different forms of a sentence, and their chatbots do not have that ability. For learning, we used deep learning approaches like them (LSTM, GRU).

We have used our dataset to train their chatbot. The same dataset has been used to compare so that the comparison can be fair. We used 100 test data in BilinBot and the others too to see the result for the English dataset. Those test data were paraphrased and were not used in the training process. We performed ROUGE evaluation on the original and predicted response. Table 1 shows the results and Fig. 7 represents the result visually. The performance of BilinBot was better than the other 2 chatbots.

TABLE I. COMPARATIVE TEST SCORES (ENGLISH DATA).

Chatbot	ROUGE-1			ROUGE-2			ROUGE-L		
	Precision	*Recall*	*F1 Score*	*Precision*	*Recall*	*F1 Score*	*Precision*	*Recall*	*F1 Score*
BilinBot	**0.96**	**0.96**	**0.96**	**0.94**	**0.94**	**0.94**	**0.96**	**0.96**	**0.96**
Conversational AI [12]	0.78	0.79	0.78	0.74	0.75	0.74	0.78	0.79	0.78
Seq2Seq AI [31]	0.72	0.73	0.72	0.68	0.69	0.68	0.72	0.73	0.72

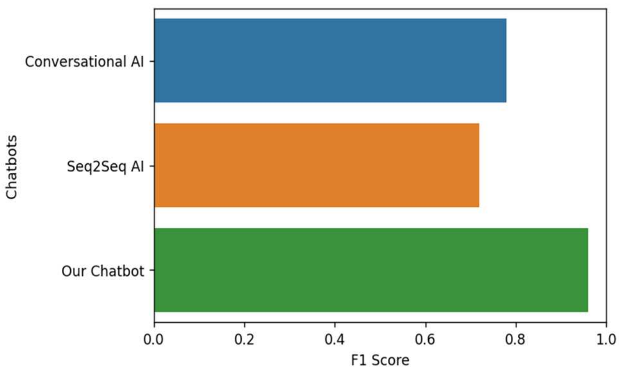

Fig. 7. Comparative test scores (English Data)

Fig. 7 shows the results for English data. The F1 score of our chatbot was 0.96 which was the highest. For Conversational AI [12] the F1 score was 0.78. For Seq2Seq AI [10] the F1 score was 0.72 which was the lowest. Our chatbot understands different paraphrased forms of sentences but the other 2 chatbots do not understand the paraphrased forms of the sentences.

Similarly, for the Bangla dataset, Table 2 shows the results and Fig. 8 illustrates the result.

TABLE II. COMPARATIVE TEST SCORES (BANGLA DATA).

Chatbot	ROUGE-1			ROUGE-2			ROUGE-L		
	Precision	*Recall*	*F1 Score*	*Precision*	*Recall*	*F1 Score*	*Precision*	*Recall*	*F1 Score*
BilinBot	**0.76**	**0.76**	**0.76**	**0.73**	**0.73**	**0.73**	**0.76**	**0.76**	**0.76**
Conversational AI [12]	0.52	0.53	0.52	0.49	0.49	0.49	0.52	0.53	0.52
Seq2Seq AI [31]	0.48	0.49	0.48	0.46	0.46	0.46	0.48	0.49	0.48

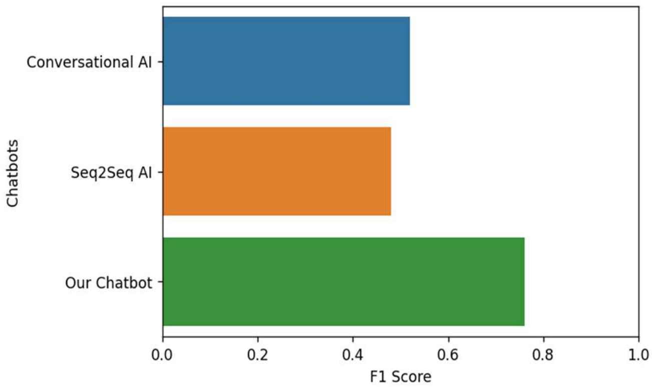

Fig. 8. Comparative test scores (Bangla Data)

VI. CONCLUSION AND FUTURE WORKS

To eliminate the language barrier, we created BilinBot, a bilingual chatbot that would assist people. In order to do that, we faced multiple challenges, such as collecting dataset for Bengali language, cleaning and processing it, and generating features using natural language processing. We have analyzed several feature engineering approaches, multiple deep learning models, and evaluated them in terms of their accuracy and performance. The best performances are given by Google BERT for feature engineering, and LSTM and GRU for the deep learning. BilinBot exhibited excellent performance and surpassed the other state-of-art work. The addition of BilinBot in our community will certainly enhance the user experience of Bangladeshi people and bridge the gap that exists due to the language barrier. It will allow business and people of all scales and classes to utilize BilinBot according to their will.

However, there are some limitations in our work that restricted us from achieving milestone. First of all, there were limited data in our dataset for the Bangla language and was bounded to only a few categories for answering questions. Therefore, as a future improvement of our work, we intend to add various other types of conversation to train the model so that it covers a wider range of sectors where it can be used. Secondly, it only works for Bangla and English language as of now. We intend to incorporate other languages in our model as well, to create a multilingual chatbot. We have designed the core framework of our chatbot, we can add a web-based user interface in the future. We can create a python module that can be imported and easily implemented in any python application.

REFERENCES

[1] A. M. Turing, "Computing machinery and intelligence," *Mind*, vol. LIX, no. 236, pp. 433–460, Oct. 1950, doi: 10.1093/MIND/LIX.236.433.

[2] J. Weizenbaum, "ELIZA—A Computer Program For the Study of Natural Language Communication Between Man And Machine," *Commun. ACM*, vol. 26, no. 1, pp. 23–28, 1983, doi: 10.1145/357980.357991.

[3] M. Dahiya, "A Tool of Conversation : Chatbot International Journal of Computer Sciences and Engineering Open Access A Tool of Conversation : Chatbot," no. May, p. 5, 2017.

[4] S. A. Abdul-Kader and D. J. Woods, "Survey on Chatbot Design Techniques in Speech Conversation Systems," *Int. J. Adv. Comput. Sci. Appl.*, vol. 6, no. 7, 2015, doi: 10.14569/IJACSA.2015.060712.

[5] R. Sharmin, S. K. Rahut, and M. R. Huq, "Bengali Spoken Digit Classification: A Deep Learning Approach Using Convolutional Neural Network," *Procedia Comput. Sci.*, vol. 171, no. 2019, pp. 1381–1388, 2020, doi: 10.1016/j.procs.2020.04.148.

[6] M. Vanjani, M. Aiken, and M. Park, "Chatbots for Multilingual Conversations," *J. Manag. Sci. Bus. Intell.*, vol. 2019, pp. 19–24, 2019, doi: 10.5281/zenodo.3264011.

[7] P. Amiri and E. Karahanna, "Chatbot use cases in the Covid-19 public health response," *J. Am. Med. Informatics Assoc.*, vol. 29, no. 5, pp. 1000–1010, 2022, doi: 10.1093/jamia/ocac014.

[8] W. Siblini, C. Pasqual, A. Lavielle, M. Challal, and C. Cauchois, "Multilingual Question Answering from Formatted Text applied to Conversational Agents," 2019, [Online]. Available: http://arxiv.org/abs/1910.04659.

[9] G. Cameron *et al.*, "Towards a chatbot for digital counselling," *HCI 2017 Digit. Make Believe - Proc. 31st Int. BCS Hum. Comput. Interact. Conf. HCI 2017*, vol. 2017-July, Jul. 2017, doi: 10.14236/EWIC/HCI2017.24.

[10] B. Setiaji and F. W. Wibowo, "Chatbot Using a Knowledge in Database: Human-to-Machine Conversation Modeling," *Proc. - Int. Conf. Intell. Syst. Model. Simulation, ISMS*, vol. 0, pp. 72–77, 2016, doi: 10.1109/ISMS.2016.53.

[11] X. Zhou *et al.*, "Multi-view response selection for human-computer conversation," *EMNLP 2016 - Conf. Empir. Methods Nat. Lang. Process. Proc.*, pp. 372–381, 2016, doi: 10.18653/v1/d16-1036.

[12] A. Ali and M. Zain Amin, "Conversational AI Chatbot Based on Encoder-Decoder Architectures with Attention Mechanism," *Artif. Intell. Festiv.*, vol. 2, no. 0, 2019.

[13] B. R. Ranoliya, N. Raghuwanshi, and S. Singh, "Chatbot for university related FAQs," *2017 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2017*, vol. 2017-Janua, pp. 1525–1530, 2017, doi: 10.1109/ICACCI.2017.8126057.

[14] R. Nallamani, "An Automated Conversation System Using Natural Language Processing (NLP) Chatbot in Python," *Cent. Asian J. Med. Nat. Sci.*, vol. 3, no. 4, pp. 314–336, 2022, [Online]. Available: http://www.centralasianstudies.orghttp//cajmns.centralasianstudies.org.

[15] R. K. Solanki, A. S. Rajawat, A. R. Gadekar, and M. E. Patil, "Building a Conversational Chatbot Using Machine Learning," pp. 285–309, 2023, doi: 10.4018/978-1-6684-7164-7.ch013.

[16] A. Vaswani *et al.*, "Attention Is All You Need," *Adv. Neural Inf. Process. Syst.*, vol. 2017-December, pp. 5999–6009, Jun. 2017, Accessed: May 16, 2023. [Online]. Available: https://arxiv.org/abs/1706.03762v5.

[17] O. Vinyals, G. Quoc, and V. Le, "A Neural Conversational Model," Jun. 2015, Accessed: May 16, 2023. [Online]. Available: https://arxiv.org/abs/1506.05869v3.

[18] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf.*, vol. 1, no. Mlm, pp. 4171–4186, 2019.

[19] K. K, Z. Wang, S. Mayhew, and D. Roth, "Cross-Lingual Ability of Multilingual BERT: An Empirical Study," pp. 1–12, 2019, [Online]. Available: http://arxiv.org/abs/1912.07840.

[20] A. Sherstinsky, "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network," *Phys. D Nonlinear Phenom.*, vol. 404, Aug. 2018, doi: 10.1016/j.physd.2019.132306.

[21] H. Deng, Y. Zhang, R. Li, C. Hu, Z. Feng, and H. Li, "Combining Residual Attention Mechanisms and Generative Adversarial Networks for Hippocampus Segmentation," *Tsinghua Sci. Technol.*, vol. 27, no. 1, pp. 68–78, Feb. 2022, doi: 10.26599/TST.2020.9010056.

[22] Z. Zhang, L. Zheng, J. Yu, Y. Li, and Z. Yu, "Three recurrent neural networks and three numerical methods for solving a repetitive motion planning scheme of redundant robot manipulators," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 3, pp. 1423–1434, 2017, doi: 10.1109/TMECH.2017.2683561.

[23] G. Chen, "A Gentle Tutorial of Recurrent Neural Network with Error Backpropagation," Oct. 2016, Accessed: May 16, 2023. [Online]. Available: https://arxiv.org/abs/1610.02583v3.

[24] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/NECO.1997.9.8.1735.

[25] B. C. Suryo Putro S, I. Wayan Mustika, O. Wahyunggoro, and H. S. Wasisto, "Improved time series prediction using LSTM neural network for smart agriculture application," *Proc. - 2019 5th Int. Conf. Sci. Technol. ICST 2019*, pp. 6–9, 2019, doi: 10.1109/ICST47872.2019.9166401.

[26] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," Dec. 2014, Accessed: May 16, 2023. [Online]. Available: https://arxiv.org/abs/1412.3555v1.

[27] B. Yue, J. Fu, and J. Liang, "Residual Recurrent Neural Networks for Learning Sequential Representations," *Inf. 2018, Vol. 9, Page 56*, vol. 9, no. 3, p. 56, Mar. 2018, doi: 10.3390/INFO9030056.

[28] Y. Li and Y. Pan, "A novel ensemble deep learning model for stock prediction based on stock prices and news," *Int. J. Data Sci. Anal.*, vol. 13, no. 2, pp. 139–149, Mar. 2022, doi: 10.1007/S41060-021-00279-9/FIGURES/11.

[29] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *1st Int. Conf. Learn. Represent. ICLR 2013 - Work. Track Proc.*, Jan. 2013, Accessed: May 16, 2023. [Online]. Available: https://arxiv.org/abs/1301.3781v3.

[30] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation Functions: Comparison of trends in Practice and Research for Deep Learning," pp. 1–20, 2018, [Online]. Available: http://arxiv.org/abs/1811.03378.

[31] A. Sojasingarayar, "Seq2Seq AI Chatbot with Attention Mechanism," no. May, pp. 1–18, 2020, [Online]. Available: http://arxiv.org/abs/2006.02767.