A Question and Answer System for Program Comprehension based on Encoder-Decoder Recurrent Neural Networks

Tin Zar Thaw
University of Computer Studies
Yangon, Myanmar
tinzarthaw@ucsy.edu.mm

Yu Yu Than
University of Computer Studies
Yangon, Myanmar
yuyuthan@ucsy.edu.mm

Si Si Mar Win
University of Computer Studies
Yangon, Myanmar
sisimarwin@ucsy.edu.mm

***Abstract*— Computer programming is an innovative cognitive tool that has transformed modern society. Two of the most widely used programming languages for web development are java and python. Therefore, source code comprehension of those programming is considered as an essential part and time-consuming task during software maintenance process. To support code comprehension process, the question and answer systems based on program comprehension have proposed. Recurrent Neural Network (RNN) based sequence-to-sequence model is one of the most commonly researched models to implement artificial intelligence question and answer system. However, it is not being applied widely in question and answer system for code comprehension. This system will learn using neural network where it uses bidirectional RNN as encoder and Luong Attention RNN as decoder. This system proposed a question and answer system to provide source code comprehension with encoder-decoder RNNs based on the code comprehension dataset: CodeQA.**

Keywords*— code comprehension, recurrent neural network, bidirectional RNN, luong attention RNN

I. INTRODUCTION

In software or web engineering, programmer uses programming languages to write the instructions that the computer will ultimately execute. Java and Python are two of the most popular object oriented programming languages for their powerful cross-platform support as well as for their extensive libraries. Java is the faster language, but Python is simpler and easier to learn. Java programming is most popular for web applications, big data and android app development. Python programming is most popular for scientific and numeric computing, machine Learning applications, image processing, language development. Although Java is simple, it has a lot of words in it, which will often leave the program with complex, lengthy sentences and explanations.

Software engineers can use the source code as their main source of information to understand the behavior of a software system through a process known as program comprehension. Everyone needs to be able to write understandable code. The algorithm and logic behind a piece of code can easily be forgotten if a programmer creates a library or other piece of code but no one can later figure out what it does. There are two basic methods that programmers can employ to increase the comprehension of their programs. The first approach involves adding documentation to code either while or after it has been written, and the second involves writing code with a focus on structure. Programmers that put a strong emphasis on structure in their code believe that subsequent maintainers will have a similar understanding of the language's syntax and mechanics. This is a reasonable assumption from time to time, but it cannot always be trusted. It is feasible to understand programs primarily written on their structure, although doing so requires the original programmer to put in a lot more effort. Programming's method functionality, purpose, property and workflow of the programming nature must be followed exactly, as well as all stylistic conventions, and everything must be consistent. A programmer must always be aware while developing code in this manner that what is obvious to increase program comprehension. This paper, a program comprehension question and answer system is proposed to understand the java and python languages' syntax and mechanics.

The Question and Answer (QA) System is an automated method that uses a context to find accurate responses to questions posed in natural language by people. QA Systems can provide a clear, concise answer. The development of deep learning and the accessibility of enormous amounts of data have allowed QA to be applied across a wide range of application areas, including news, science, film, the medical industry, and others. This paper focus on QA system for java and python source code comprehension. QA-based source code comprehension is the ability to answer questions about functionality, purpose, property, and workflow of programming nature. Neural network algorithms have become popular in intelligent QA models due to recent advances in artificial intelligence. There are most commonly used types of neural networks are feedforward neural networks, convolutional neural networks and recurrent neural networks. Recurrent Neural Networks have offered superior performance for except correlation between answers and questions for QA systems [8]. So, this paper intends to propose a source code comprehension QA system using encoder-decoder RNNs.

The rest of the paper is structured as follows. Related words is discussed in Section II and Section III presents background theory. Section IV describes the dataset and Section V explains the implementation of the proposed system and experimental results. Section VI concludes the proposed system and directs for future work.

II. RELATED WORKS

A QA system gives a direct answer to a frequently asked question in natural language. There are two types of QA systems: open domain and closed domain. Closed domain system is the task of answering questions from a particular narrow domain and offer answers on specific topics, while open-domain systems are based on general ontologies and broad unrestricted knowledge. In this research, the closed domain QA system for code comprehension is presented. Computer programming languages has a central role in our society because every AI devices and AI software's are not created without computer programming like java, python,

2023 IEEE Conference on Computer Applications (ICCA) | 978-1-6654-3599-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCA51723.2023.10182053

javascript and etc. Many researchers have proposed different approaches to deal with program or code comprehension for software development and education purpose.

The authors [4] have proposed a QA system that can automatically answer to questions about the python programming language based on a structured knowledge base. The QA System presented reasonable answers to questions about Python and automatically populated PythonQAS knowledge repository.

The authors explored the field of QA with respect to an educational context and analyzed existing frameworks [9]. A framework was been built for a dynamic self-evolving Concept Network, built specifically for a given subject, chapter, lecture etc. The architecture had three modules: the Dynamic Concept Network, the Question Analysis and the Answer Retrieval. The model was been compared with the existing BiDAF and Omnibase START models based on 50 QA pairs. The model was able to correctly answer 80% of the definition based questions and 65% of the other type of questions. Mover, the proposed system answers accurately than other models.

The authors [5] explored the possibility of building a tutorial question answering system for Java programming from 106,386 questions sampled from a community-based question answering forum. The system intended to support building a tutorial QA system and to support the creation of a dataset. Retrieval-based, generative and a hybrid models was been implemented on the given dataset and retrieval-based models obtained high recall rates. Generative models answered that the responses were small and did not completely answered the question.

The authors [3] investigated a QA system on introductory Java programming using the transformer model to prove to be a base for supplementary education tools. They trained on QA pairs from the online programming forum Stack Overflow with five Transformer models. Each model was evaluated using perplexity as an automatic metric and a qualitative evaluation done by the author. The models was high quality models by measuring by the automatic metric evaluation. However, the qualitative evaluation showed that the generated responses was short, generic, repetitive, and even contradicting. They concluded that the transformer model trained on Stack Overflow data could not answer introductory Java programming concepts.

The authors [7] created a dataset with 3636 reading comprehension QA pairs based on a transformer-based deep neural network model to obtain convenient answers for the Bangla Education system. Deep neural network architectures: Long Short-Term Memory (LSTM), ELECTRA, Bidirectional LSTM with attention, Recurrent Neural Network, and Bidirectional Encoder Representations from Transformers was used and the trained model of BERT performed a satisfactory outcome with 87.78% of testing accuracy and 99% training accuracy, and ELECTRA provided training and testing accuracy of 82.5% and 93%, respectively.

This study mainly focuses on a QA system in code comprehension, which answer java and python questions accurately. The proposed system mainly differs from the recent studies by adapting QA system from encoder-decoder RNN to java and python code complementation.

III. BACKGROUND THEORY

The encoder-decoder model is a way of using recurrent neural networks (RNN) for sequence-to-sequence prediction problems. It involves two recurrent neural networks: encoder and decoder [2]. Gated Recurrent Unit (GRU) is basic cell of RNN and it uses less memory for less training parameters. GRU trains faster than LSTM's whereas LSTM is more accurate on dataset utilizing longer sequence [6]. An encoder takes the input sequence and encapsulates them as the internal state vectors. A decoder uses the internal state vectors to generate the target sequence. This system proposed a QA system to provide code comprehension based on encoder-decoder RNNs with GRU and CodeQA. The proposed system uses keras and tensorflow to build Encoder-Decoder RNNs.

A. Encoder

The proposed system is used Bidirectional RNN (BRNN) as encoder and describes the algorithm process of Bidirectional RNN as follows in [10].

Inputs:

- List of words for each sentences in the batch
- Hidden states: no of layer, directions, batch size and hidden size.

Bidirectional RNN algorithm:

- Convert word indexes to embedding.
- Pack batch of sequences for RNN module.
- Forward and Backward pass through RNN cells.
- Unpack padding.
- Sum bidirectional outputs.
- Return output and final hidden states.

Outputs:

- Output features from the last hidden layer of the RNN cell and updated hidden states.

B. Attention

Luong attention uses the top hidden layer states as well as Bahdanau attention uses the concatenation of the forward and backward source hidden states from the top hidden layer. While Luong is well-suited for both as uni-directional and only accepting the top layer outputs, Bahdanau's focus is the combination of a uni-directional encoder and a bi-directional decoder. The Luong attention model is simpler than Bahdanau attention. If Luong attention detects the decoder's hidden state at time t, it will calculate the alignment score, create a context vector, concatenate it with the decoder's hidden state, and then predict.

Luong has three different types of alignment functions: dot, general and concat according to equations (1) to (3). Dot function calculates the alignment score by multiplying the hidden states of the decoder and the hidden state of the encoder at time T. Like dot function, general alignment function multiplies two hidden states with a weight matrix. Concat alignment function adds the hidden states of the decoder and encoder before going through a linear layer. Then, a tanh activation function is applied on the output before being multiplied by a shared weight matrix.

$$Score(h_t, h_{\bar{s}}) for\ dot = h_s^T * h_s^T \quad (1)$$

$$Score(h_t, h_s^-) for\ general = h_s^T * W_a * h_s^T \quad (2)$$

$$Score(h_t, h_s^-) for\ concat = h_a^T \tanh(W_a[h_t + h_s^T]) \quad (3)$$

All the hidden states of the encoder are taken by the global attention model to calculate the context vector (Wa). A variable length alignment vector (a_t) is the size of the number of time steps in the source sequence that is inferred by comparing the current target hidden state (h_t) with each of the source hidden state (h_s).

C. Decoder

This system is used Luong attention RNN as decoder and describes the algorithm process of decoder as follows:

Inputs:

- One time step of input sequence batch.
- Encoder model's luong inputs.

Luong RNN:

- Get installing of current info word.
- Forward through unidirectional RNN cell.
- Calculate consideration loads from the current cell yield.
- Multiply consideration loads to encoder yields to get new "weighted entirety" setting vector.
- Concatenate weighted setting vector and cell yield.
- Predict next word.
- Return output and last shrouded state.

Outputs:

- Probabilities of each word being the correct next word in the decoded sequence and final hidden state of RNN cell.

IV. DATASET DESCRIPTION

CodeQA[1] is a free form QA dataset for comprehension of the source code and it contains two datasets: java with 119,778 QA pairs and python with 70,085 QA pairs. This dataset can serve as a useful research benchmark for source code comprehension. Due to the varied nature of programming code comments, CodeQA covers a wide range of information contained in code, from methods to variables. These sets of data can be grouped into four categories: functionality, purpose, property, and workflow of programming nature. These four categories are asked by using eight question types. In this system, the QA system for programming language Source Code Comprehension with Recurrent Neural Networks. These dataset are divided into three parts for training, validation and testing according to the table 1. The example QA pairs of python dataset are shown in the table II.

TABLE I. CODEQA DATASET DESCRIPTION

	question-answer pairs	Training (60%)	Validation (20%)	Testing (20%)
Java	119,778	71866	23956	23956
Python	70,085	42051	14017	14017

TABLE II. SAMPLE QA PAIRS OF PYTHON DATASET

Type	Example Questions and Answers
Functionality	Question: What does a utility generator pad? Answer : Argument list and Dictionary values.
Purpose	Question: For what purpose does flatpage object? Answer : For the current site.
Property	Question: When do this function use? Answer : When threads are being used.
Workflow	Question: What does the code return in a json blob? Answer : some basic document info

V. IMPLEMENTION OF THE PROPOSED SYSTEM AND EXPERIMENTAL RESULT

This section explains the proposed methodology of QA system illustrated in Fig. 1 in details. The proposed system takes two datasets as input and preprocess these datasets. Preprocessing change over the Unicode strings to ASCII. Next, all letters are changed to lowercase and trimmed all non-letter characters with the exception of essential accentuation. The vocabulary list keeps a mapping from words to lists including all words in a sentences of the dataset and RNN and counts number of unique words of the dataset. After preprocessing steps, the QA system encodes the preprocessed dataset with Bidirectional RNN (BRNN) and encodes the output of decoder with three luong attention alignment functions: dot, general and concat. Finally, the system produced six models based on java and python datasets. To evaluate six models, Bilingual Evaluation Understudy (BLEU) metric is used to measure the quality of models' answer with testing datasets according to the equations 4 and 5.

$$BLEU = \min(1, \frac{hypothesis_{length}}{Reference_{length}}) * Accuracy \quad (4)$$

$$Accuracy = \text{Max number of words occurs in reference/}$$
$$\text{Total no of words in hypothesis} \quad (5)$$

Where, $hypothesis_{length}$ means the QA systems answer length and $reference_{length}$ is the real or expect answer in the dataset.

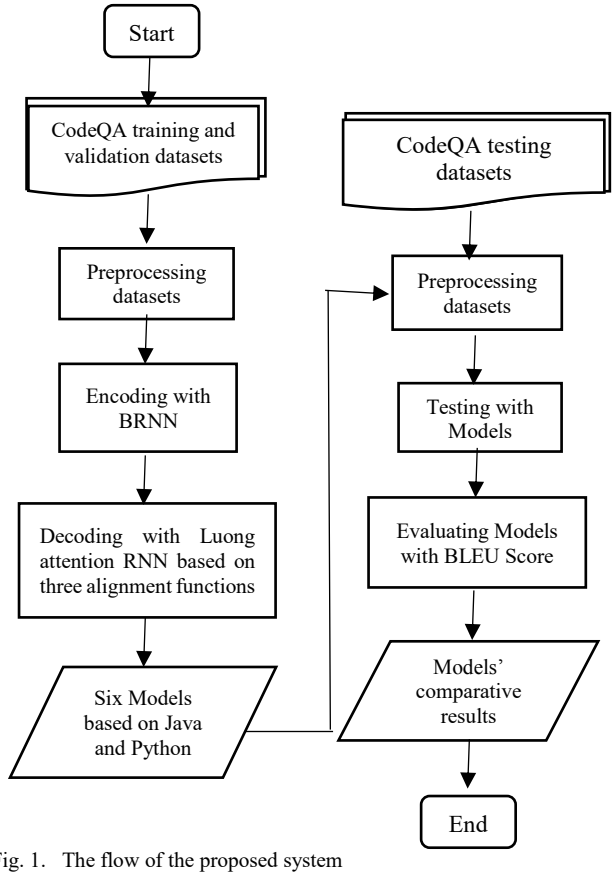

Fig. 1. The flow of the proposed system

TensorFlow [7] is an open-source machine learning library to implement the models. The QA system for Program

or Code Comprehension is proposed on Java and Python Programming Languages.

A. Hyperparameter Tuning

When the proposed system is implementing with tensorflow, hyperparameters play a crucial role as it controls RNN behavior during the training process. So, the suitable hyperparameters are needed to choose for the sequence to Sequence Models. In this system, the validation datasets are used to choose the length of input sequence, number of encoder and decoder layers and batch size using the general models of python and java datasets. The default hyperparameters are set as follows:

- Learning rate: 0.001
- Hidden size: 500
- Batch size: 16
- Dropout: 0.1
- Minimum sentence length: 2
- Encoder hidden layer : 4
- Decoder hidden layer: 4
- Checkpoint Iteration: 3000

Firstly, the possible three different input sequence lengths are 30, 50 and 70 are considered using above default hyperparameters because the python's question answer pairs are short and summary answers. The general python model is created by using the default hyperparameters and by changing input sequence lengths based on RNN. According to the experimental BLEU score result, the appropriate input sentence length of python dataset is 30. Although 50 and 70 lengths has produced the similar BLEU scores with 30, the processing times are larger than the appropriate input sentence length. The appropriate input sentence length of java dataset is 70. The other two length has produced for lower BLEU score value because the length of java question answer pairs are long than python pairs. Therefore, the default sentence length are set as follows:

- Input sentence length for python : 30
- Input sentence length for java : 70

Secondly, the possible three batch size are consider as: 16, 32 and 64.The general python model is created and test with validation dataset and the result found that 64 is the most suitable for the RNN model. So, batch size is set as follows:

- Batch size: 16

Finally, the number of encoder and decoder layers is tuned to get the suitable value among 2, 3 and 4. In this system, the same layer of encoder and decoder is used to train the models with difference layer numbers. These numbers have not produced significant BLEU score results but processing time is longer as increasing layer number. So, the two layers for each of encoder and decoder are the most suitable according to the experimental results. The final hidden layer is set with two.

- Encoder hidden layer : 2
- Decoder hidden layer : 2

B. Experimental Result

The code complementation QA system are been implemented sequence to sequence model of RNN to answer right information and adapt self-learning of java and python languages using tuning hyperparameters.

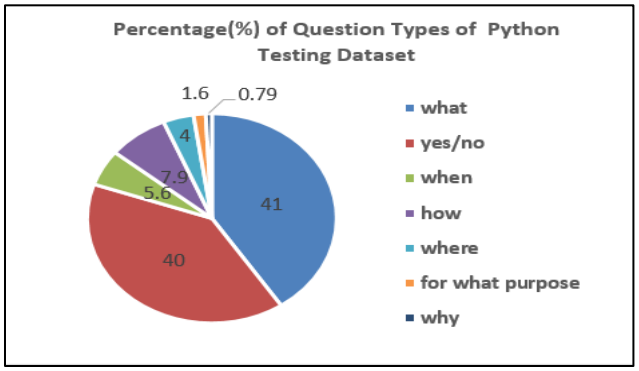

Fig. 2. The percentage of Seven Question Types of Python Testing Dataset.

Fig. 2 describes the percentage of seven question types of Python testing dataset. The two question types: "what" and "yes/no" are 41% and 40% of the testing python data. The other question types are under the percentage of ten.

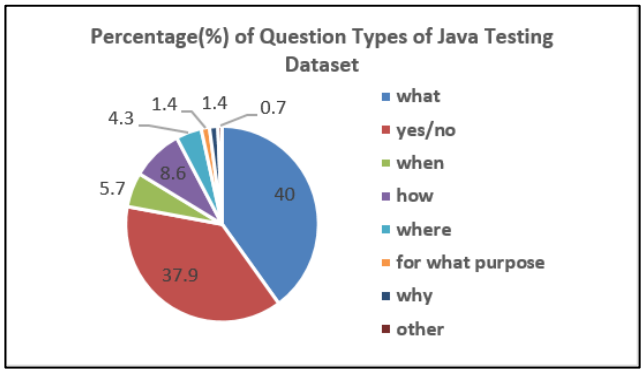

Fig. 3. The percentage of Seven Question Types of Java Testing Dataset.

Fig. 3 describes the percentage of eight question types of Java testing dataset. The two question types: "what" and "yes/no" are 40% and 37.9% of the testing python data. The other question types are under the percentage of ten.

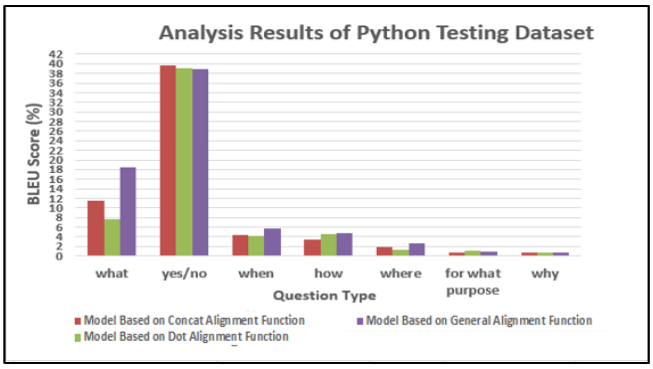

Fig. 4. BLEU Score results of three models based on Luong attention alignment functions: concat, dot and general for python testing dataset.

According to the python analysis results, three models can answer the "yes/no" question type correctly in Fig. 4. For python QA pairs, BLEU Score result of question type "what" are difference. Although the percentage of this question type contains 41% of test data, the models produces BLEU Scores with 18.4% of general function, 11.5% of concat function and 7.7% of dot function correctly. The other question types can be answered by the model based on general alignment function with more BLEU Score.

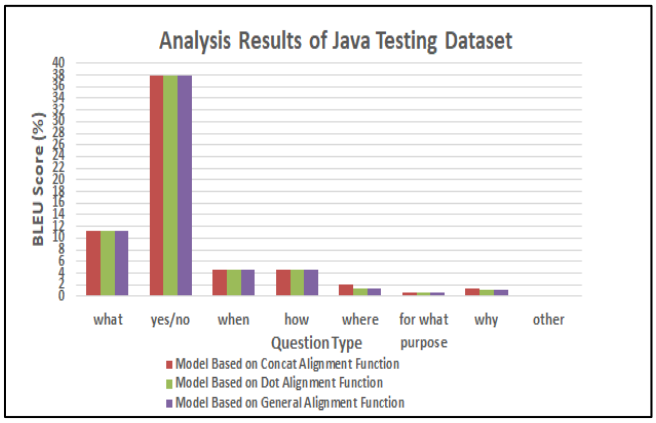

Fig. 5. BLEU Score results of three models based on Luong attention alignment functions: concat, dot and general for java testing datase.

According to Fig. 5, three models can answer the "yes/no" question type with full percentage and the "what" question type can answer the percentage 11.1 out of 37.9 correctly. The ''when'' and how question types show similar BLEU Scores for three models as it answered most of the questions.

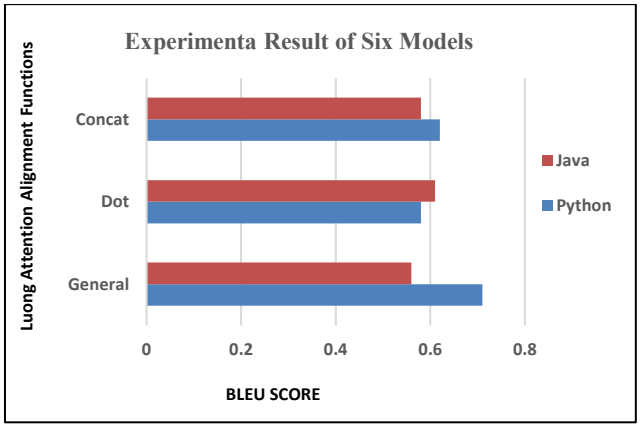

Fig. 6. The experiment result of encoder-decoder RNN models

The proposed system will learn using RNNs where bidirectional RNN one is used as encoder and Luong attention RNN is used as decoder to produced six models. Fig. 6 explained the experiment results of six models based on three luong attention alignment functions and two datasets. Python models has higher BLEU Scores than java models because of the same checkpoint iteration and different size of datasets. Java dataset needs more iteration than python according to CodeQA dataset. For python dataset, the model based on general alignment function gets the highest BLEU Score and this model is the most suitable for the python dataset. For java dataset, answers of model based on dot alignment function are more accurate than other models.

VI. CONCLUSION AND FUTURE WORK

The code comprehension QA system has been implemented by using encoder-decoder RNNs to answer right information and adapt self-learning based on CodeQA dataset. The proposed system will learn using neural networks where bidirectional RNN one is used as encoder and Luong Attention RNN is used as decoder. According to three Luong Attention Alignment functions: dot, general and concat, this system created six code comprehension QA models based on java and python languages. This system evaluates six models' performance using BLEU score. According to the experimental results, the accuracy of python models are higher BLEU Scores than the java models because of same checkpoint iteration and difference dataset size. The accuracy of models are difference on the nature of the dataset QA pairs. Python general model is the best for python languages and Java dot model is the best for java languages. Generally, if the model can be trained with increasing number of iterations in model training, the more accurate QA system for program or code comprehension for programming languages.

In this study, LSTM RNN cells and Bahdanau attention are not considered. As future work, it would be interesting to compare the LSTM and GRU RNNs using two attentions: Luong and Bahdanau.

REFERENCES

[1] Chenxiao Liu and Xiaojun Wan, "CodeQA: A Question Answering Dataset for Source Code Comprehension Empirical Methods in Natural Language Processing, EMNLP 2021.

[2] Jakub Silka, Michal Wieczorek and Marcin Wozniak, "Recurrent neural network model for high-speed train vibration prediction form time series", Neural Computing and Applicationsx 29 January, 2022.

[3] Lukas Szerszen and Richard James Glassey "Question answering on introductory Java programming concepts using the Transformer", Degree Project In Computer Sceence and Engineering, School of Electrical Engineering and Computer Science (EECS), 2021.

[4] Marcos Ramos, Maria Varanda Pereira, Pedro Rangel Henriques, "A QA System for learning Python", Federated Conference on Computer Science and Information Systems, 2017.

[5] Mayank Kulkarni and Kristy Boyer," Toward Data-Driven Tutorial Question Answering with Deep Learning Conversational Models", Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications,june,2018.

[6] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural net-works. Signal Processing, IEEE Transactions on, 45(11):2673–2681, 1997.

[7] Mumenunnessa Keya, Abu Kaisar Mohammad Masum, Sheikh Abujar, Bhaskar Majumdar and Syed Akhter Hossain, "Bengali Question Answering System Using Seq2Seq Learning Based on General Knowledge Dataset", 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2020.

[8] Oumayma Chergui, Ahlame Begdouri and Dominique Groux-Lecle," Integrating a Bayesian semantic similarity approach into CBR for knowledge reuse in Community Question Answering", Knowledge-Based Systems, Volume 185, 1 December 2019, 104919.

[9] Walaa A. Elnozahya, Ghada A. El Khayata, Lilia Cheniti-Belcadhib and Bilal Saidc, "EDUQA: Educational Domain Question Answering System Using Conceptual Network Mapping", 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

[10] "TensorFlow," Available at https://www.tensorflow.org/.