

Semantic Improvement of Question-Answering System with Multi-Layer LSTM

Hanif Bhuiyan[1(✉)], Md. Abdul Karim[2], Faria Benta Karim[3], and Jinat Ara[4]

[1] Data61, CSIRO, CARRS-Q, Queensland University of Technology, Queensland, Australia
hanif.bhuiyan@data61.csiro.au
[2] Department of Computer Science and Engineering,University of Asia Pacific, Dhaka, Bangladesh
[3] Department of Electrical and Electronic Engineering, University of Asia Pacific, Dhaka, Bangladesh
[4] Department of Computer Science and Engineering, Jahangirnagar University, Savar, Dhaka, Bangladesh

Abstract. Question answering (QA) is a popular method to extract information from a large pool of data. However, effective feature extraction and semantic understanding, and interaction between question and answer pairs are the main challenges for the QA model. Mitigating these issues, here, we propose a Long-Short-Term-Memory (LSTM) model using the recurrent neural network. This model aims to extract the effective interaction between questions-answers pairs using the Softmax and Max-pooling function. The experiment performs on a publicly available Wiki-QA dataset to identify the effectiveness of the proposed model. The evaluation performs by comparing the result with other existing models. The comparison shows that the proposed approach is significant and competitive for QA and achieves 83% training and 81% testing accuracy on the Wiki-QA dataset.

Keywords: Question answering · Long-short-term-memory · Recurrent neural network · Information extraction

1 Introduction

An intelligent question-answer (QA) is a popular human machine interaction approach to satisfy the user by retrieving correct answers or information of a specific question [1]. The rapid increase of web data allows a web user to store data and make it available for the community for future use [2,3]. However, this massive storage data makes the searching process is a difficult task [4]. This difficulty allows NLP researchers to develop new tools such as Question Answer to find the correct information from structured and unstructured corpus data [5].

© Springer Nature Switzerland AG 2021
O. Gervasi et al. (Eds.): ICCSA 2021, LNCS 12951, pp. 338–352, 2021.
https://doi.org/10.1007/978-3-030-86970-0_24

In the past few years, artificial intelligence and knowledge base methods were popular in question-answer systems for many NLP researchers [6,7]. These methods were performed through several text analysis algorithms to analyze and identify the most similar keywords [8] for considering the appropriate answer to the question [9]. Surprisingly, these methods have a lack of semantic meaning. The accuracy of the QA system mostly depends on the QA sentence embedding and mapping. It helps to generate a sequential representation of the context. Generally, sequentially QA sentence mapping is difficult and hard to get good accuracy. Besides, it addresses the shortcoming of knowledge-based techniques [10]. However, a fully functional question-answer system is relatively challenging,and several researchers have tried to build a more systematic process to improve accuracy [11,12].

In recent days, to minimize the shortcoming of knowledge-based techniques, deep learning provides several methods and techniques [12]. The majority of NLP solutions have replaced by deep learning techniques to improve accuracy and produce accurate results (especially the abstract question-answering approaches) [13]. This method performs through an artificial neural network and showed significant improvement in the question-answer system. Therefore, deep learning uses widely and accepted by the NLP researcher to discover the distributed data representation process [14]. Among several deep learning methods, recurrent neural network (RNN) is a significant innovation and frequently used method to map the question and answer sentences [11]. It performs through continuously identifying the meaning of each word of a sentence to represent the entire meaning of the sentences and mapping with the other sentence word.

Recently researchers take advantage of recurrent neural network models for semantic analysis of contextual data [12,15–18]. Feng et al. [15] and Wang and Nyberg [16] used recurrent neural networks and Long Short-Term Memory Networks. These works are limited to identify short text semantics of small corpus data. They didn't consider predicting the full-sentence semantics of corpus data, resulting in a poor semantic understanding. Zhang and Ma [19] proposed a deep learning-based QA model with a single LSTM network concept. They didn't consider a multi-layer LSTM network that might reduce accuracy and produces inaccurate results.Although these methodologies are effective in QA systems,the implemented techniques still bring low accuracy.

In this paper, we implement a multilayer LSTM network through RNN architecture for the single sentence question-answer domain. This work aims to solve the need for semantic understanding and improve the overall accuracy of corpus data. Besides, an attention mechanism implements to improve the semantic understanding of the proposed model. It helps to increases the interaction between the question-answers pairs. An optimization technique uses to improve the performance of the proposed model. A comprehensive experiment conducted to determine the proposed model's effectiveness on the publicly available WikiQA data set.The experiment result shows that neural network-based multilayer LSTM architecture is more effective for question-answer tasks than single LSTM architecture. The comparison result of existing model considers evaluating the

proposed model. The compared result shows that the proposed model is significant and achieved higher accuracy than the existing system (83% and 81% accuracy for training and testing, respectively). The contributions of this paper summarized as follows:

(a) Multilayer LSTM recurrent neural network architecture is implemented to identify the semantic meaning and effectively map the input question-answer sentence vector.

(b) The proposed model incorporates the ReLU activation function and softmax attention mechanism to obtain the most semantic representation of the QA pair.

(c) Max-pooling and Adam optimizer reconciled to calculate the accuracy for improving the accuracy rate.

The rest of the paper presents as follows. In Sect. 2, represent a brief overview of related work. In Sect. 3, presents the proposed LSTM-RNN architecture for the question answering framework. In Sect. 4, added the details about the implementation procedure of the proposed QA model. In Sect. 5, conclude the experimental result with performance analysis. Finally, a detailed conclusion or discussion presents in Sect. 6.

2 Related Work

To understand the insight of corpus data and retrieving the desired outcome from a massive number of semantic web data, QA is a popular approach. There are several techniques to perform QA tasks [6,7,11,15,16]. Larson et al. [20] proposed work enhances the importance of POS tagging and tf-idf technique in the question-answering system. Sharma et al. [21] proposed a QA model considering artificial intelligence algorithms. It shows the significant importance of the AI technique in QA approach. Both of these techniques are promising, however these fail to represent semantic understanding effectively and resulting low accuracy. Therefore, NLP researcher has started to implement deep learning methods for structure and unstructured data analysis. Motivated by deep learning techniques, Cai et al. [22] implement a neural network structure using a stacked Bidirectional Long Short-Term Memory (BiLSTM) model for question-answer tasks. Rohit et al. [23] implemented a neural network architecture using a single-layer LSTM architecture and memory network for closed domain question-answer analysis. The experimental result of these two proposed approaches [22,23] addressed the necessity of neural network architecture in the QA approach. Though these works showed impressive results, they only considered single layer LSTM architecture and didn't focus on the attention mechanism and optimization technique. Less consideration of the LSTM layer, attention mechanism, and optimization technique could lead to a lack of semantic understanding and reduce overall accuracy.

Therefore, Li et al. [26] emphasized the importance of multi-layer LSTM architecture. Furthermore, Yin et al. [18] added the importance and significance of the neural network technique or architecture to improve the Question Answering system. Here, we implement four LSTM architecture layers using the recurrent neural network (RNN) architecture with an attention mechanism and optimization algorithm to extract higher semantic accuracy.

3 Question Answer (QA) Framework

In this section, we describe the proposed deep learning-based question answering system (shows in Fig. 1). The proposed architecture design into three phases: input processing and embedding phase, LSTM neural network phase, and QA feature mapping phase. At first, the input processing and embedding phase performs the question and answer pair's vector representation. Then, a multilayer LSTM neural network implements to extracts the hidden feature of the input question-answer sentence. It acts as an encoder through the activation function (ReLU). Next, the attention mechanism (Softmax) considers for QA feature mapping. Max-Pooling uses to identify the higher semantic feature. Adam

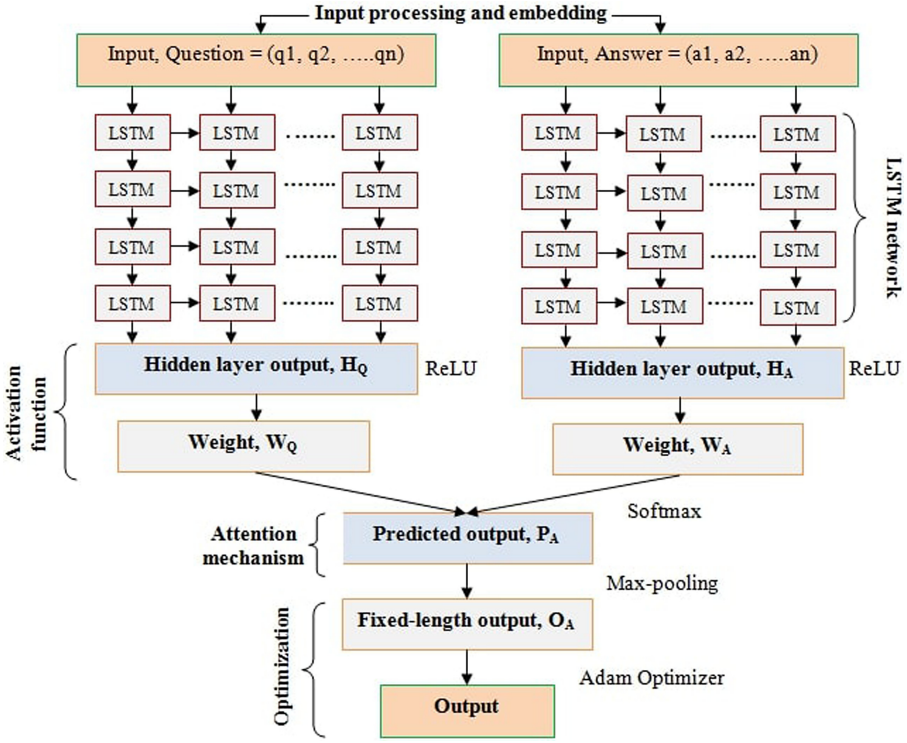


Fig. 1. Proposed LSTM question answer neural network architecture

optimizer applies for probability distribution to calculate the accuracy of the proposed model.

3.1 Input Processing and Embedding

To automatic question answering system, a set of questions and answers is taken as input first. Here, Question set, Q = (q_1, q_2, q_3,.....q_m) and Answer set, A = (a_1, a_2, a_3,....a_n), where m is the length of the question and n is the length of the answer. After feeding the sets of questions and answers into the embedding layer, run four layers of the LSTM network over the questions and answer embedding to obtain their hidden state vector. Mathematical formula of hidden state vector denoted by H_Q and H_A where H_Q is the question hidden state and H_A is the answer hidden state shown in Eqs. 1 and 2.

$$H_Q = [h_{q1}, h_{q2}, ..., h_{qn}] \subset R^{d*n} \tag{1}$$

$$H_A = [h_{a1}, h_{a2}, ..., h_{am}] \subset R^{d*m} \tag{2}$$

3.2 Long Short Term Memory (LSTM) Network

Long short term memory (LSTM) is a modified form of a recurrent neural network (RNN). Hochreiter and Schmidhuber [24] developed it to introduce a memory cell structure to classify, process, and make a prediction of data. Generally, the LSTM network performs using three control gates (shown in Fig. 2) : input gate (i_t), forget gate (f_t), and output gate (O_t).
The input gate allows incoming data or signal and determines which input information (x_t) of the network has to remember or save for the current cell state, C_t, and added to the neuron state. In the input layer, the sigmoid function determines the necessary updating of the information, and the tanh function generates vector C_t as shown in Eqs. 3 , 4 , 5.

$$C_t = f_t * C_{t-1} + i_t * C_t \tag{3}$$

$$i_t = Sigma(W_i.[h_{t-1}, x_t] + b_i) \tag{4}$$

$$C_t = tanh(W_c.[h_{t-1}, x_t] + b_c) \tag{5}$$

Then, forget gate modulate states and allow how many cell states have to remember or forget its previous state for the current neuron state C_{t-1}. It only saves the verified data (by the algorithm) and removes the unsatisfied information through the forget gate. The forgetting probability calculation shows in Eq. 6, where h_{t-1} represents previous neuron output, and X_t represents current neuron input.

$$f_t = Sigma(W_f.[h_{t-1}, X_t] + b_f) \tag{6}$$

Finally, the output gate, O_t determines the amount of cell state C_t that has to transmit in the current output h_t, which shows in Eq. 7 and 8, where σ is a

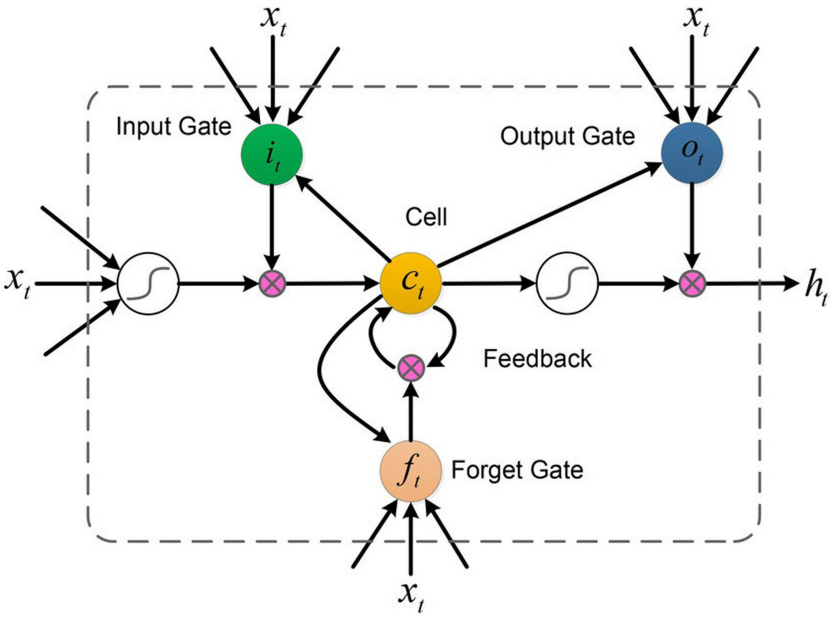


Fig. 2. LSTM basic cell architecture [18]

logistic sigmoid function, x_t is t-t^h word vector of the sentence, and h_t refers to hidden state.

$$O_t = Sigma(W_o.[h_{t-1}, x_t] + b_o) \tag{7}$$

$$h_t = O_t * tanh(C_t) \tag{8}$$

Moreover, here we define four layer forward LSTM network where input sequence of layer A, is $x_A = (x_{A1}, x_{A2},\ldots\ldots, x_{An})$ which produce the output sequence, $y_A = (y_{A1}, y_{A2},\ldots\ldots, y_{An})$. The output of layer A become the input of layer B and produce output sequence, $y_B = (y_{B1}, y_{B2},\ldots\ldots, y_{Bn})$. Similarly output of layer B considered as the input of layer C and produce output sequence, $y_C = (y_{C1}, y_{C2},\ldots\ldots, y_{Cn})$. Finally output of layer C considered as input of layer D and produce the forward hidden sequence $h_t = (h_1, h_2, h_3,\ldots\ldots h_n)$. The four layer forward LSTM structure is shown in Fig. 3.

4 QA Feature Mapping and Sequencing

To feature mapping, and QA pair sequencing, activation function, attention mechanism, and optimization technique considers and describes through the following subsections. The activation function considers extracting the appropriate feature of question-answer pairs. Then the attention mechanism uses for semantically mapping question-answer tokens. Finally, an optimization algorithm increases the overall accuracy. A detailed overview is described through the following subsections.

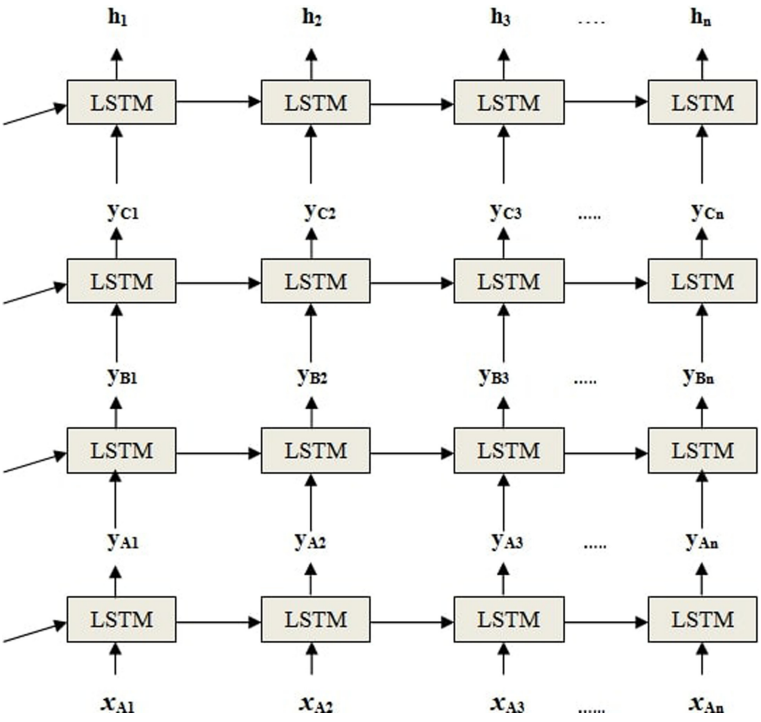


Fig. 3. Forward LSTM network architecture

4.1 Activation Function

In terms of the proposed model, we applied the activation function (showed in Fig. 1) to determine the necessary information which should add or delete and pass the selective information to the next hidden state. To determine whether information passes or not, the proposed model uses the Rectified Linear Unit[1] (ReLU) activation function on the output of hidden layer neurons, which is popular in deep learning techniques (shown in Eq. 9 , 10). The ReLU function provides a real number output between [0, 1]. It represents the weight of the information. It only active the neuron if the result of the linear transformation is greater than 0. It refers that for a negative input value, it produces the result 0 that indicates the deactivated status of the neuron. Generally, the ReLU function activates a certain number of neurons simultaneously. So it is more efficient than the sigmoid and tanh function and provides better performance by reducing the negative sampling or result.

$$F(x) = max(0, x) \tag{9}$$

$$R(z) = 1, z > 0; 0, z \leq 0 \tag{10}$$

[1] https://en.wikipedia.org/wiki/Rectifier_(neural_networks).

4.2 Attention Mechanism

In the proposed model, the attention mechanism reduces the amount of data loss of the LSTM layer. The attention mechanism is appropriate for inferring the mapping of different data (especially for question-answer tokens). Here, a popular attention mechanism known as a Softmax function[2] applies to mapping information from the questions and answers words to make answer predictions. Softmax is the most effective attention mechanism in dealing with probability distribution problems. The calculation process of the Softmax function shows in Eq. 11.

$$Softmax(x_{\mathrm{i}}) = exp\frac{(x_{\mathrm{i}})}{\sum(x_{\mathrm{j}})} \tag{11}$$

4.3 Optimization

To increase the accuracy of the proposed model, here we implement max-pooling technique and the Adam algorithm. Maximum pooling uses to identify the highest weight and transform the input into a fixed-length output. Max-pooling is the most conventional pooling method to calculate the weight of arbitrary information [25]. Max-pooling is denoted by Eq. 12, where M is the size of the pooling region or context.

$$Output(i) = max(x_{\mathrm{j}}); 1 \leq j \leq M * M \tag{12}$$

Besides, Adam optimizer[3] implements to improve the performance of the proposed model. It is a well-known optimization algorithm developed by Diederik Kingma and Jimmy Ba for predicting the outcome of a large scale of data. Generally, it uses natural language and computer vision problems to improve performance by continuously updating the parameters.

5 Implementation Details

This section describes the implementation process of the neural network model in detail. The implementation process performs into the data set, pre-processing, training, and LSTM network. Here we train the model and observe each phase of testing accuracy rate to identify the highest accuracy of the model.

5.1 Dataset

In this step, we introduce a public data set Wiki-QA Data set with its source, characteristics, number of pairs in detail. Here, the Wikipedia Question Answer

[2] https://en.wikipedia.org/wiki/Softmax_function.
[3] https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/.

dataset[4] is considered for the experiment. Carnegie Mellon University and the University of Pittsburgh launched this data set in 2010 for academic research. This Wiki-QA dataset includes 1165 pairs of questions and answers with both incorrect and missing/null answers. This Wiki-QA dataset focus on different types of question and answer, for instance, single supporting fact, yes/no, personal information, historical information, time, numbers, etc. We used this data set for both training and testing purposes (considering 80% for training and 20% for testing purposes). The experiment performs on this data set to evaluate the performance and effectiveness of the proposed model.

5.2 Preprocessing

The Wiki-QA dataset is a natural language question-answer dataset with numbers of notation, missing values, and unformed structure. To convert this unstructured data into a suitable data structure and make it human-readable, pre-processing performs as follows:

Removing Notation: At first, we remove the HTML and punctuation tags from the obtained text or string. Then all the single characters and spaces have been eliminated.

Splitting: Next, splitting the sentences into questions and answers to feed into the proposed model.

Tokenize: The third step is pre-processing for tokenizing the sentences into different words. For performing tokenizing, we used the Keras tokenizer. For example, a sample question: Was Alessandro Volta a professor of chemistry?. The tokenize form of this sample question is: [“Alessandro”, “Volta”, “a”, “professor”, “of”, “chemistry”, “?”].

Indexing: To make the equal length of questions and answer sentences, we perform post-padding and post-truncation operations as various length questions and answers introduced in the dataset. The ambiguous question-answer length ratio might lead to poor results. Therefore, the maximum length of the question and the answer converts into 14 and 30 (<30 might reduce accuracy and >30 might introduce over-fitting problem) respectively to increase accuracy and minimize the over-fitting problem. Those QA pair tokens that exceed this range will automatically eliminate.

5.3 Training Details

Here, 80% of the Wiki-QA dataset is used for the training phase. It helps to identify the optimal architecture of the proposed model. The proposed model trains until 100 epochs. For training, the batch size sets to 100. The memory

[4] www.cs.cmu.edu/~ark/QAdata/data/Question_Answer_Dataset_v1.2.tar.gz.

size is limited to the first 100 sentences. Padding operation performs to minimize the fixed-length problems to obtain fixed-size data. The embedding of the null symbol replaces with zero. For the final answer prediction, Softmax uses in the memory layer. Sometimes model performance was not significant. Therefore, we repeated the process with different initialization or epoch (30, 50, and 100) to identify the highest and lowest accuracy (shown in Fig. 5).

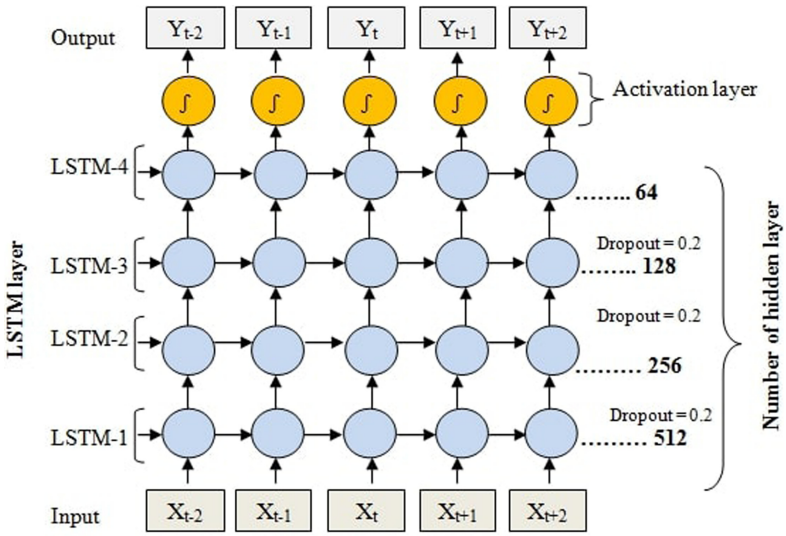


Fig. 4. Basic structure of multi layer forward LSTM-RNN architecture

5.4 LSTM Network

A LSTM recurrent neural network architecture considers here to implement the proposed QA model (shows in Fig. 1). The basic LSTM recurrent neural network architecture shows in Fig. 4. To train the proposed QA model, we used the TensorFlow[5] platform using the Keras[6] library.

At first, the word-embedding layer uses for both questions and answer encoding. Then LSTM recurrent neural network layer is used for passing the encoded question where the number of dense layers is considered 512, 256, 128, and 64 for each LSTM memory network block. To increase the accuracy of the proposed architecture dropout layer is used. For training the LSTM memory network, 0.2 dropouts are used after the word embedding layer to avoid the over-fitting problem. The LSTM memory network was trained for 100 epochs with Adam optimizer and identifies the highest training and testing accuracy around 83%

[5] https://www.tensorflow.org/.
[6] https://keras.io/.

and 81%, respectively. Finally, the activation layer considers for passing the true-positive information and predicting the final fixed-length output.

6 Performance Analysis

In this section, we experimented proposed QA model on the Wiki-QA data set to observe the performance and identify the effectiveness.The experiment has conducted through training the LSTM recurrent neural network with an attention mechanism and optimization algorithm. Besides, multiple layers of LSTM architecture considered increasing the performance of the proposed model. For semantically representing the question-answer pairs, we used the attention mechanism and optimization technique that makes the proposed model convincing.

The proposed QA approach validates through considering the different range of epochs numbers on the Wiki-QA data set. To find a stable result, we gradually increase the epoch number (number of iteration about 100 times) from 0 to 100 to reveal the best accuracy.

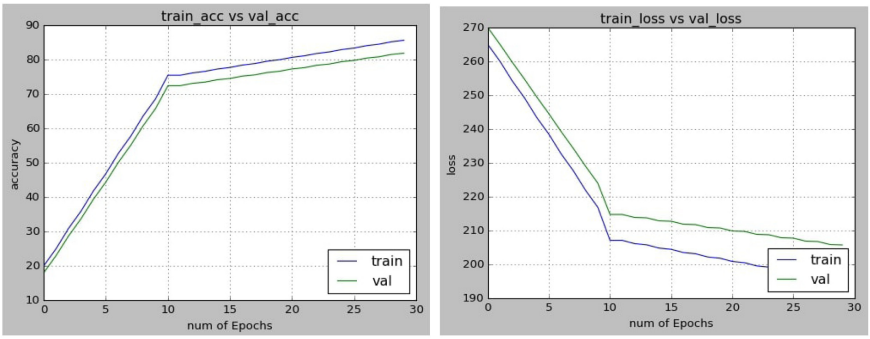
Fig. 5 shows that, in the proposed model, changes in the epoch numbers (10 to 100) can increase or decrease the loss function and overall accuracy. However, in both training and testing, as the number of iteration has increased, the loss in training and testing has decreased (Fig. 5(a–c)). Therefore, a certain number of iteration or epochs could reduce the loss percentages and increase the performance.

Here, Table 1 shows the overall accuracy of the Wiki-QA data set for both training and testing phases. It shows that in 30 epochs for the training and testing phase, the proposed model achieves 78.56% and 76.20% accuracy, respectively. Similarly, the accuracy of both training and testing has increased by around 2% in 50 epochs. In contrast, the proposed model shows significant improvement in both training and testing accuracy of 83.62% and 81.90% in the 100 epoch range, which is the maximum for the Wiki-QA data set. It depicts that a certain number of iteration could able to increase accuracy and improve the final results.

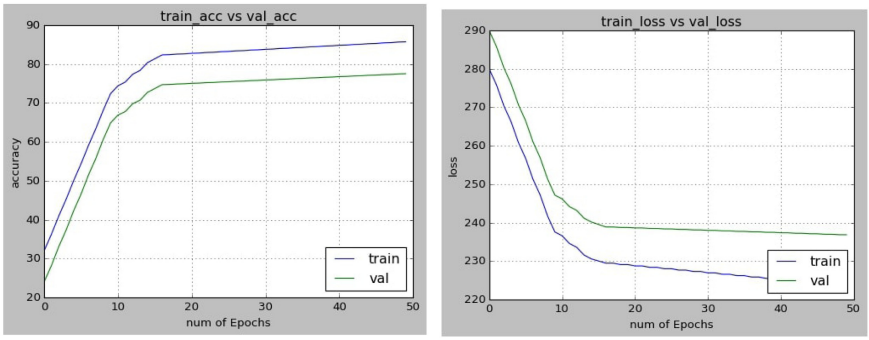
Table 1. Overall accuracy of the proposed QA model considering different range of epoch number

No of epoch	Accuracy (Train)	Accuracy (Test)
30	78.56%	76.20%
50	80.79%	78.12%
100	83.62%	81.90%

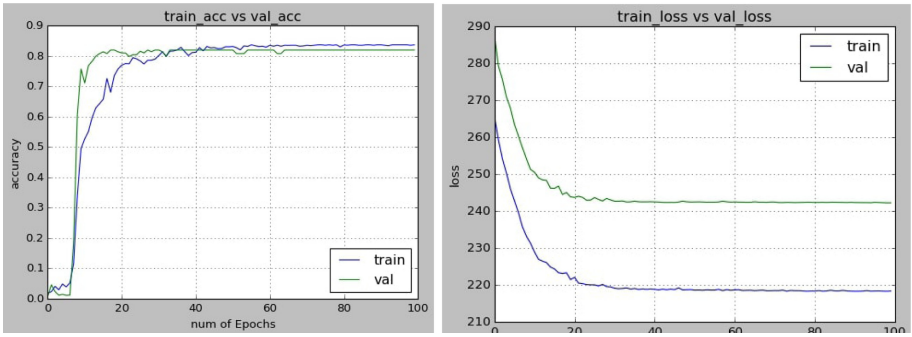
To evaluate the proposed approach's effectiveness, we conclude the compared result of the proposed QA model with existing popular QA models. According to the multi-layer prototype, the proposed model compared with Li et al. [26] which proposed a multi-layer LSTM-based QA model without using attention



(a) Training & Testing accuracy vs. loss for 30 epochs



(b) Training & Testing accuracy vs. loss for 50 epochs



(c) Training & Testing accuracy vs. loss for 100 epochs

Fig. 5. Training and testing loss with accuracy according to epoch size 100 (a–c)

mechanism and achieve 37.11% training and 37.67% testing accuracy. Comparing this result with proposed model result, it emphasizes that multi-layer LSTM architecture with attention mechanism produces a better result than without attention mechanism. Through an attention mechanism, each word of the QA pair gets more weight. Besides, an attention mechanism represents the seman-

tic meaning or relation among multiple words without any external conditions. Similarly, the proposed model compared with Haigh et al. [12] introduced a QA model using match-LSTM with pointer net and achieve 60.30% training and 45.00% testing accuracy. This comparison result shows that the match-LSTM with the pointer net QA model has less accuracy than our proposed four layers LSTM model with the attention mechanism. Finally, to evaluate the neural network architecture, proposed model compared with Tan et al. [8] that proposed a LSTM-CNN architecture for retrieving answer to the question and achieving overall 67.20% training and 65.70% testing accuracy. Comparing the result of this work with the result of the proposed LSTM-RNN architecture depicts that RNN architecture is more reliable than CNN architecture in QA problem. RNN architecture helps to increase the significant performance of the QA model rather than CNN architecture.

However, comparing the proposed QA approach with existing models it could be speculated that the presented model results are superior to the most existing models. The accuracy of the proposed model is relatively higher than the accuracy of the addressed existing models. Our proposed model achieves 83% (Training) and 81% (Testing) accuracy. Besides, the experimental result shows that the proposed model is effective for semantic feature representation and semantic feature mapping. Moreover, performance analysis and evaluation results depict that the proposed model has excellent performance, significant improvement, and effectiveness in an intelligent question answering system.

7 Conclusion and Future Work

In this paper, we proposed a deep learning-based Question Answer (QA) approach. The proposed system aims to identify the answer to a particular question. Here, four layers of LSTM architecture implements through the recurrent neural network with an attention mechanism for improving semantic understanding and representation of the QA tokens. The experiment result proves the efficiency of the proposed approach. Also, it shows a significant improvement compared with other solutions. In this work, the experiment is performed in the Wiki-QA dataset and compared with some existing system only. In future, our goal is to evaluate our proposed model through large-scale datasets such as SQuAD, SemEval-cQA, and SciQA, respectively and consider more existing approaches to evaluate the proposed approach widely.

References

1. Diefenbach, D., Both, A., Singh, K., Maret, P.: Towards a question answering system over the semantic web. Seman. Web **11**(3), 421–439 (2018). https://doi.org/10.3233/SW-190343
2. Yu, W., et al.: A technical question answering system with transfer learning. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 92–99 (2020). https://doi.org/10.18653/v1/2020.emnlp-demos.13

3. Bhuiyan, H., Oh, K.J., Hong, M.D., Jo, G.S.: An effective approach to generate Wikipedia infobox of movie domain using semi-structured data. J. Internet Comput. Serv. **18**(3), 49–61 (2017)
4. Ara, J., Bhuiyan, H.: Semantic recognition of web structure to retrieve relevant documents from Google by formulating index term. Adv. Comput. Commun. Comput. Sci. (2019). https://doi.org/10.1007/978-981-13-6861-5_43
5. Bouziane, A., Bouchiha, D., Doumi, N., Malki, M.: Question answering systems: survey and trends. Procedia Comput. Sci. **73**, 366–375 (2015). https://doi.org/10.1016/j.procs.2015.12.005
6. Saxena, A., Tripathi, A., Talukdar, P.: Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 4498–4507 (2020). https://doi.org/10.18653/v1/2020.acl-main.412
7. Banerjee, P., Baral, C.: Knowledge fusion and semantic knowledge ranking for open domain question answering. arXiv preprint arXiv:2004.03101 (2020)
8. Ara, J., Bhuiyan, H.: Upgrading YouTube video search by generating tags through semantic analysis of contextual data. In: Uddin, M.S., Bansal, J.C. (eds.) Proceedings of International Joint Conference on Computational Intelligence. AIS, pp. 425–437. Springer, Singapore (2020). https://doi.org/10.1007/978-981-15-3607-6_34
9. Pradel, C., Sileo, D., Rodrigo, Á., Peñas, A., Agirre, E.: Question answering when knowledge bases are incomplete. In: Arampatzis, A., et al. (eds.) CLEF 2020. LNCS, vol. 12260, pp. 43–54. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58219-7_4
10. Sadhuram, M.V., Soni, A.: Natural language processing based new approach to design factoid question answering system. In: 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), pp. 276–281. IEEE (2020). https://doi.org/10.1109/ICIRCA48905.2020.9182972
11. Gao, S., Chen, X., Ren, Z., Zhao, D., Yan, R., et al.: Meaningful answer generation of e-commerce question-answering. arXiv preprint arXiv:2011.07307 (2021)
12. Haigh, A., van de Graaf, C., Rachleff, J.: Extractive question answering using match-LSTM and answer pointer
13. Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., Gao, J. et al.: Deep learning based text classification: a comprehensive review. arXiv preprint arXiv:2004.03705 (2021)
14. Abbasiyantaeb, Z., Momtazi, S.: Text-based question answering from information retrieval and deep neural network perspectives: a survey. arXiv preprint arXiv:2002.06612 (2021)
15. Feng, M., Xiang, B., Glass, M.R., Wang, L., Zhou, B., et al.: Applying deep learning to answer selection: a study and an open task. In: 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), pp. 813–820. IEEE (2015). https://doi.org/10.1109/ASRU.2015.7404872
16. Wang, D., Nyberg, E.: A long short-term memory model for answer sentence selection in question answering. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pp. 707–712 (2015)
17. Tan, M., Santos, C.D., Xiang, B., Zhou, B.: LSTM-based deep learning models for non-factoid answer selection. arXiv preprint arXiv:1511.04108 (2015)
18. Yin, W., Yu, M., Xiang, B., Zhou, B., Schütze, H., et al.: Simple question answering by attentive convolutional neural network. arXiv preprint arXiv:1606.03391 (2016)

19. Zhang, L., Ma, L.: Coattention based bilstm for answer selection. In: 2017 IEEE International Conference on Information and Automation (ICIA), pp. 1005–1011. IEEE (2017). https://doi.org/10.1109/ICInfA.2017.8079049
20. Larson, T., Gong, J.H., Daniel, J.: Providing a simple question answering system by mapping questions to questions
21. Sharma, Y., Gupta, S.: Deep learning approaches for question answering system. Procedia Comput. Sci. **132**, 785–794 (2018). https://doi.org/10.1016/j.procs.2018.05.090
22. Cai, L., Zhou, S., Yan, X., Yuan, R.: A stacked BiLSTM neural network based on coattention mechanism for question answering. Comput. Intell. Neurosci. (2019). https://doi.org/10.1155/2019/9543490
23. Rohit, G., Dharamshi, E.G., Subramanyam, N.: Approaches to question answering using LSTM and memory networks. In: Bansal, J.C., Das, K.N., Nagar, A., Deep, K., Ojha, A.K. (eds.) Soft Computing for Problem Solving. AISC, vol. 816, pp. 199–209. Springer, Singapore (2019). https://doi.org/10.1007/978-981-13-1592-3_15
24. Yildirim, Ö.: A novel wavelet sequence based on deep bidirectional LSTM network model for ECG signal classification. Comput. Biol. Med. **96**, 189–202 (2018). https://doi.org/10.1016/j.compbiomed.2018.03.016
25. Zhao, Q., Lyu, S., Zhang, B., Feng, W.: Multiactivation pooling method in convolutional neural networks for image recognition. Wireless Commun. Mobile Comput. (2018). https://doi.org/10.1155/2018/8196906
26. Li, Y.: Two layers LSTM with attention for multi-choice question answering in exams. In: IOP Conference Series: Materials Science and Engineering, vol. 323, no. 1, p. 012023. IOP Publishing (2018)