Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

Question Answering System Based on Bideirectional Long-Short-Term Memory (Bilstm)

Sahar Sami Raheem Katib[1], Mohammed Hasan Abdulameer[2]
([1]computer science college of education University of kufa, Najaf , 54003, Iraq). edu@uokufa.edu.iq
([2] faculty of education for women, university of kufa, Najaf, 54003 Iraq). edu_girls@uokufa.edu.iq

[1] sahars.alkatib@student.uokufa.edu.iq
[2] mohammed.almayali@uokufa.edu.iq

https://doi.org/10.46649/fjiece.v3.2.9a.18.5.2024

Abstract. *In the modern world, Q&A systems are essential for promoting better communication between people and technology. These systems play an important role in collecting information quickly and efficiently, and this leads to great progress in learning, teaching and development in many areas of life. Using deep learning techniques, this research addresses the problem of excellent prediction of the questions that need to be answered. We created a question-and-answer system using "Bidirectional Long Short-Term Memory (BiLSTM)", a modern neural network known for its accuracy and results in text analysis and natural language understanding. This technique is more effective in understanding questions and producing very accurate answers because of its special ability to pay attention to preceding and following information in a sentence. Preprocessing was used to remove unnecessary, unimportant and time-consuming data. The "Stanford Question Answering Dataset version 2 (SQuAD 2.0)" was used, which is considered one of the important datasets used in the field of machine learning and natural language processing. The following evaluation metrics were used to evaluate the model's performance: "Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), Recall, Precision, Loss, F1 Score, and Exact Match (EM). The results, based on 150 epochs (EPOGs) and 128 batch sizes with a cleaned dataset split into 70% training and 30% test/validation (15% each), are as follows: "Precision (0.966), Loss (0.591), F1-score (0.966), Recall (0.967), EM (0.967), MRR (0.918), MAP (0.776), and accuracy (0.966)". Interestingly, the highest performance was observed when using the accuracy measure.*

Keywords: *Natural language processing, Question answering system, deep learning, LSTM, BiLSTM.*

1. INTRODUCTION

Addressing questions within natural language processing poses a formidable challenge, demanding machines to comprehend specific text passages and generate accurate responses [1]. In the realm of question-answering systems, various deep learning applications have emerged, including Bidirectional Encoder Representations from Transformers (BERT), A Lite BERT (ALBERT), QANet, Generative Pre-trained Transformer (GPT), DrQA, Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (BiLSTM).[2]. Our focus, rooted in the efficacy of BiLSTM, stems from its contextual understanding, ability to overcome vanishing gradient issues, robust representation of time, effectiveness in natural language analysis, and seamless integration with other techniques [3]. This paper explores the integration of BiLSTM into question-answering systems to enhance their performance by effectively processing and understanding the intricacies of human language.

As a background to our research, many pioneering studies have contributed that have made significant progress in the field of question-response systems, and we will mention some of them:

1. **BLSTM + Attention Model** ([4]): Song et al. utilized bidirectional long-term memory (BLSTM) combined with attention mechanisms, showing a significant improvement in performance metrics on Big Data, achieving a MAP score of 71.99% and an MRR score of 80.01%. While the model demonstrates excellent accuracy, its computational demand may limit scalability in resource-constrained environments.
2. **Deep Neural Networks for Answer Selection** ([5]): Zhang and Peng proposed a model using word vectors to represent questions and answers, with LSTM networks evaluating their similarity. The model, tested on the InsuranceQA dataset, reached an accuracy of 85%. The strength of this approach lies in its effective use of vector space representations, though it may struggle with semantic nuances not captured purely by vector similarities.
3. **CN-BiLSTM on bAbi Dataset** ([6]): Li et al. introduced a CN-BiLSTM model that incorporates a multi-channel information collection framework, achieving a high prediction accuracy of 99.3% on the training set and 94% on the test set. The model excels in handling structured data but may need further refinement for unstructured data types found in natural language.
4. **BERT Algorithm in Turkish Question Answering** ([7]): GEMİRTER and GOULARAS developed a system utilizing the BERT algorithm for Turkish, assessed with SQuAD (Tr) and NewsQA (Tr). It achieved an Exact Match of 55.26% and an F-Score of 67.07%. The model's strength lies in its language-specific adaptation, though its performance varies significantly with the complexity of the questions.
5. **BERT-based System with Co-Attention and Self-Attention** ([8]): Yin's study incorporated co-attention and self-attention mechanisms in a BERT-based framework. The system was tested across multiple datasets, yielding an average EM of 46.56 and F1 score of 58.90. This approach enhances interpretability of model decisions but requires fine-tuning to improve accuracy.
6. **Deep Learning and Dynamic Memory Model** ([9]): Antony and Paul emphasized a dynamic memory model to address complex tasks in a Tamil language-based system. Although the second iteration of the model showed improvement, limitations in handling unknown data point to a need for enhanced generalizability.

In light of these studies which collectively underscore the rapid advancements and diverse methodologies within the field, there remain challenges, particularly in improving systems' ability to generalize across various languages and data types, as well as managing the computational demands of increasingly complex models. Against this backdrop, our paper offers valuable insights and results concerning question-answer systems, leveraging deep learning techniques. This paper specifically proposes the development of a model for a question and answer system utilizing BiLSTM technology. The proposed model aims to harness the sequential data processing capabilities of BiLSTMs to enhance understanding and response accuracy in question-answering scenarios, addressing some of the current limitations faced by existing systems in dealing with complex query contexts and varied data structures. The remainder of this paper is organized as follows: The proposed method is detailed in Section 2, followed by the presentation of results in Section 3. Finally, the conclusion is discussed in Section 4.

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

2. THE PROPOSED METHODOLOGY

2.1. Theoretical part

In this part, it is necessary to understand the theoretical foundations of LSTM and BiLSTM by clarifying both the algorithmic framework and the structural nuances of these techniques. This foundational understanding is crucial before we begin to define the proposed methodology for building a question-and-answer system that leverages BiLSTM technology.

1. 1 LSTM

Long short-term memory networks (LSTM) addresses the Vanishing Gradient Problem inherent in deep networks by incorporating specialized modules known as "gates." These gateways are necessary to control the flow of information by allowing or blocking its passage [10] .

LSTM Algorithm:

Initialization:

- Initialize the cell state C_0 and the hidden state H_0 to zero.

Parameters:

- Weight matrices W_f, W_i, W_c, W_o for the forget gate, input gate, cell update, and output gate.
- Bias vectors b_f, b_i, b_c, b_o for the forget gate, input gate, cell update, and output gate.
- Recurrent weight matrices U_f, U_i, U_c, U_o from the previous hidden state to the forget gate, input gate, cell update, and output gate.

For each time step t from 1 to T:

1. **Forget gate:** $f_t = \sigma(W_f \cdot x_t + U_f \cdot h_{t-1} + b_f)$
2. **Input gate:** $i_t = \sigma(W_i \cdot x_t + U_i \cdot h_{t-1} + b_i)$
3. **Cell update:** $\tilde{c}_t = \tanh(W_c \cdot x_t + U_c \cdot h_{t-1} + b_c)$
4. **Cell state update:** $C_t = f_t * C_{t-1} + i_t * \tilde{c}_t$
5. **Output gate:** $o_t = \sigma(W_o \cdot x_t + U_o \cdot h_{t-1} + b_o)$
6. **Hidden state update:** $h_t = o_t * \tanh(C_t)$

The structure of the LSTM cell is visually represented in Figure 1 [11].

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

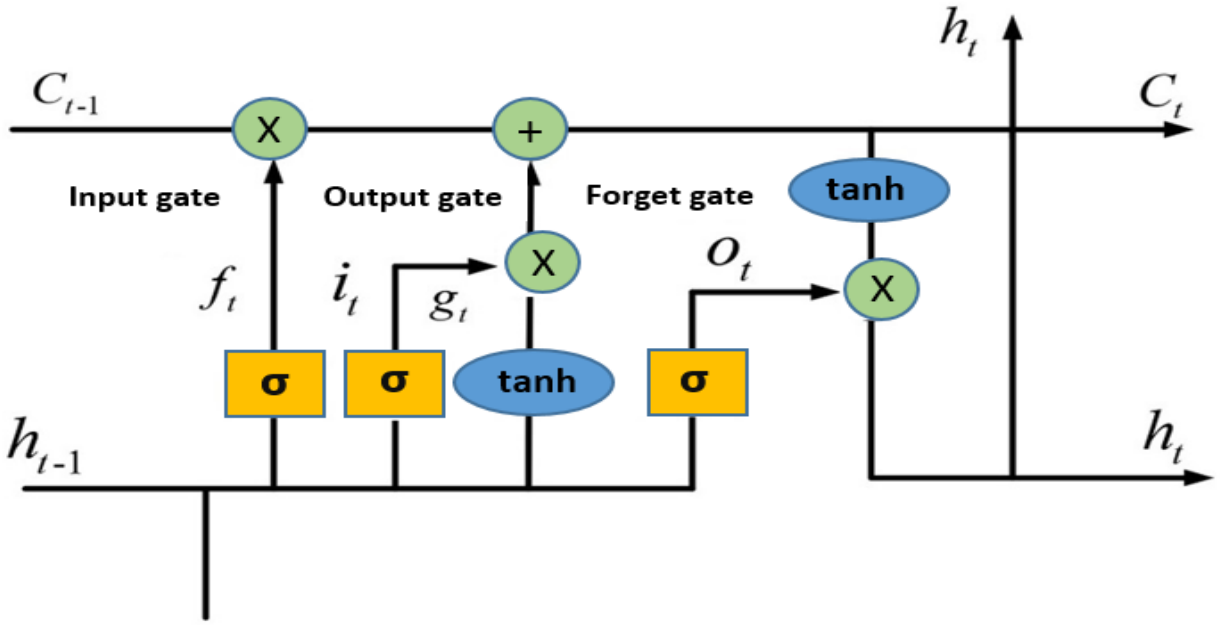

Fig. 1. The Architecture of an LSTM Cell. [11]

2. BiLSTM

Bidirectional LSTM (BiLSTM) extends the concept of LSTM by introducing an additional direction of data flow. Unlike traditional LSTM, which depends solely on the preceding context, BiLSTM incorporates both the preceding and succeeding contexts, enhancing its ability to grasp the overall context. Figure 2 [11] visually depicts the architecture of BiLSTM, and its algorithm is elucidated as follows:

BiLSTM Algorithm:

Initialization:

- Initialize the cell states C_0^{fwd}, C_0^{bwd} and the hidden states H_0^{fwd}, H_0^{bwd} to zero for both the forward and backward LSTM components.

Parameters:

- For the forward LSTM: Weight matrices $W_{fwd}^f, W_{fwd}^i, W_{fwd}^c, W_{fwd}^o$, bias vectors $b_{fwd}^f, b_{fwd}^i, b_{fwd}^c, b_{fwd}^o$, and recurrent weight matrices $U_{fwd}^f, U_{fwd}^i, U_{fwd}^c, U_{fwd}^o$.
- For the backward LSTM: Weight matrices $W_{bwd}^f, W_{bwd}^i, W_{bwd}^c, W_{bwd}^o$, bias vectors $b_{bwd}^f, b_{bwd}^i, b_{bwd}^c, b_{bwd}^o$, and recurrent weight matrices $U_{bwd}^f, U_{bwd}^i, U_{bwd}^c, U_{bwd}^o$.

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

Processing:

1. **Forward LSTM Pass:**
   - Process each timestep t from 1 to T using the LSTM equations:
     - **Forget gate:** $f_t^{fwd} = \sigma(W_{fwd}^f \cdot x_t + U_{fwd}^f \cdot h_{t-1}^{fwd} + b_{fwd}^f)$
     - **Input gate:** $i_t^{fwd} = \sigma(W_{fwd}^i \cdot x_t + U_{fwd}^i \cdot h_{t-1}^{fwd} + b_{fwd}^i)$
     - **Cell update:** $\tilde{c}_t^{fwd} = \tanh(W_{fwd}^c \cdot x_t + U_{fwd}^c \cdot h_{t-1}^{fwd} + b_{fwd}^c)$
     - **Cell state update:** $C_t^{fwd} = f_t^{fwd} * C_{t-1}^{fwd} + i_t^{fwd} * \tilde{c}_t^{fwd}$
     - **Output gate:** $o_t^{fwd} = \sigma(W_{fwd}^o \cdot x_t + U_{fwd}^o \cdot h_{t-1}^{fwd} + b_{fwd}^o)$
     - **Hidden state update:** $h_t^{fwd} = o_t^{fwd} * \tanh(C_t^{fwd})$
2. **Backward LSTM Pass:**
   - Process each timestep t from T to 1 in reverse order:
     - Follow similar LSTM equations but with backward-specific parameters, updating h_t^{bwd} and C_t^{bwd} at each step.
3. **Output Combination:**
   - For each timestep t, combine the outputs of the forward LSTM h_t^{fwd} and backward LSTM h_t^{bwd} typically by concatenation to form the final output $h_t = [h_t^{fwd}, h_t^{bwd}]$.

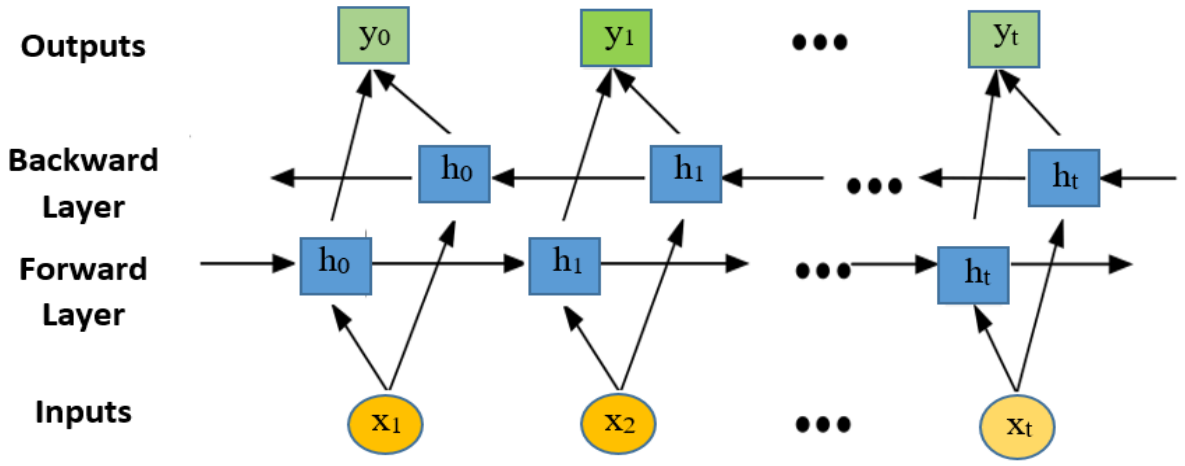

Fig. 2. The BiLSTM Structure [11]

2.2. Proposed method

The proposed methodology for designing a question and answer system based on deep learning techniques, specifically utilizing BiLSTM technology, is illustrated in Figure 3. This figure encapsulates all the key stages involved in crafting the integrated model for the question and answer system. The methodology section will elaborate on the initial two stages, encompassing pre-processing and model construction. Subsequently, the ensuing stages of training, evaluation, and prediction will be detailed in the results section for a thorough understanding of the system's development and efficacy.

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

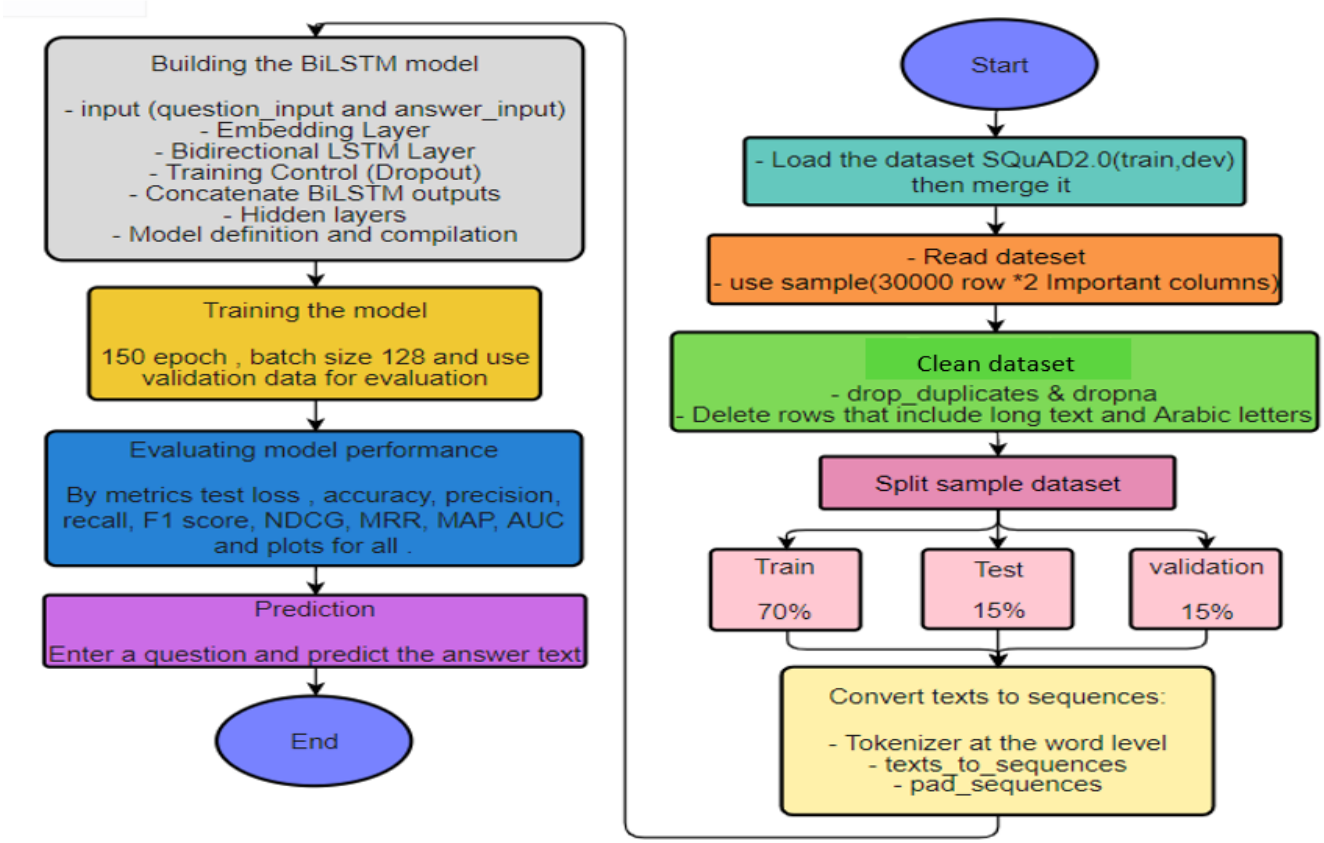

Fig. 3. The Proposed Methodology

1. *pre-processing stage*

In the initial step of dataset pre-processing, we commenced by downloading the SQuAD 2.0 dataset, which was available in JSON format and comprised two distinct segments (training and development). The subsequent actions involved merging these two segments, transforming the file extension into a CSV format, and extracting a representative sample for utilization in our study. Subsequently, we applied several treatments to the sampled data, such as eliminating redundant fields, lengthy texts, and empty fields containing Arabic letters. Following these treatments, the dataset was partitioned into three subsets, constituting 70% for training, and 15% each for validation and testing. Transitioning to the second stage of pre-processing, the textual content underwent segmentation into word-based units using a tokenizer. The texts were then converted to numeric sequences with zeros at the end to make these sequences equal in length using sequence conversion and padding techniques. This process resulted in numeric strings of equal length, supplemented with zeros as needed. The culminating dataset, post the dual pre-processing stages, is visually represented in Table 1 and Table 2 below.

Table 1. The Dataset After the First Stage of Pre-Processing

Question	answer_ text
When did Beyoncé start becoming popular?	in the late 1990s
What areas did Beyoncé compete in when she was growing up?	singing and dancing

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

Table 2. The Dataset After the Second Stage of Pre-Processing

Question	answer_ text
11,160,29,4311,18,520,5,3376,938,3,371,447,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0	503,1560,85,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2,6,1,582,3377,2070,4,1,791,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0	2070,3,9169,23454,23455,1948,722,4,1515,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

To illustrate all the steps and operations that we performed during the pre-processing stage, this can be done through the following Figure 4.

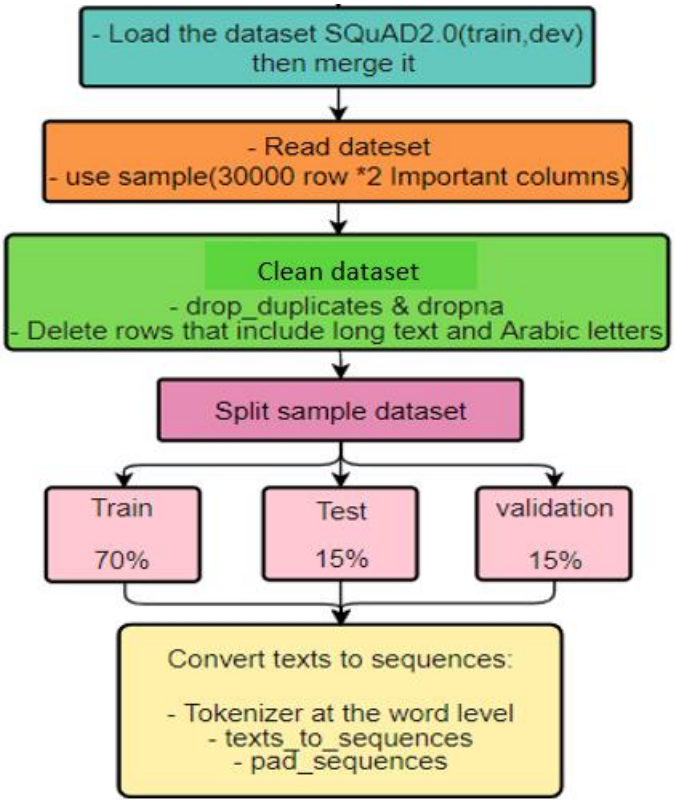

Fig.4. Pre-Processing Steps

2. *The proposed model BiLSTM*

This section provides a comprehensive explanation of the algorithm employed in constructing the proposed model. It covers the steps, layers, and equations involved in the model building process. It is crucial to note that the BiLSTM technique is utilized, and its algorithm has been previously discussed.

Illustrating all the main steps and operations conducted during the model building phase, this is depicted in Figure 5.

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

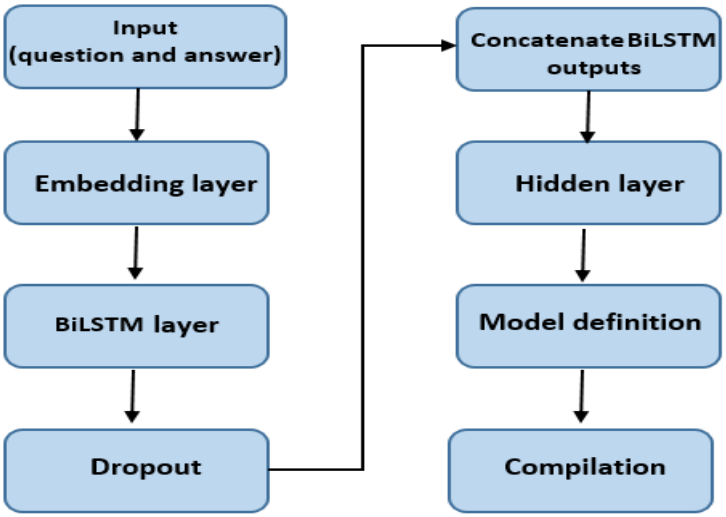

Fig. 5. Illustrates the Primary Steps Involved in Constructing the Model.

Algorithm for BiLSTM-based Question Answering System

1. **Define the Model Parameters:**
   - **Embedding Dimensions (**embedding_dim**)**: The size of each word vector.
   - **LSTM Units (**lstm_units**)**: The number of units in each LSTM layer.
   - **Vocabulary Size (**vocab_size**)**: The total number of unique words in the training corpus.
   - **Maximum Sequence Length (**max_length**)**: The fixed length of input sequences.

2. Input Definition:

- Define two input layers, question_input and answer_input, both shaped as (max_length,) to hold the sequences of word indices for questions and answers.

3. Embedding Layer Setup:

- **Embedding Matrix (**E**)**: Initialize an embedding layer to transform integer indices into dense vectors. The matrix size is (vocab_size, embedding_dim).
- **Input Sequences (**X**)**: Represent input sequences for questions and answers.
- **Embedded Output (**X_embed**)**: Compute embedded representations as $X_{embed}=E \cdot X$, where each input word is replaced by its corresponding vector.

4. Prepare the Embedding Layer:

- Apply the embedding layer to both question_input and answer_input to obtain dense vector representations, Q_embed and A_embed.

5. Define LSTM Layers:

- Initialize a Bidirectional LSTM layer to capture forward and backward contextual information from the input sequences. The layer concatenates the outputs from both directions at each time step.

6. Process Inputs Through LSTM:

- Pass the embedded question (Q_embed) and answer (A_embed) sequences through the BiLSTM layer to obtain sequential hidden states, Q_LSTM and A_LSTM.

7. Add Dropout to Avoid Overfitting:

- Apply dropout regularization to both LSTM outputs (Q_LSTM and A_LSTM) to prevent overfitting, resulting in Q_LSTM_dropout and A_LSTM_dropout.

8. Merge Outputs:

Concatenate the dropout-treated LSTM outputs along the last axis to integrate information from both questions and answers into a unified representation, Merged.

9. Add Additional Dropout after Merging:

- Apply another dropout layer to the merged output (Merged) to further enhance model generalization.

10. Define Hidden Layers:

- Construct multiple dense layers with ReLU activation functions to capture complex patterns in the data. Apply dropout after each hidden layer for added regularization.
- The ReLU activation function is defined as RReLU(x)=max(0,x).

11. Define the Final Output Layer:

- Set up an output layer with a softmax activation function to predict the probability distribution over possible answers. The softmax function for the ii-th element is defined as $Softmax(zi)=\frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}}$

12. Compile the Model:

- Create a Model object with inputs as question_input and answer_input, and the final output as Output.
- Compile the model using the Adam optimizer and sparse categorical cross-entropy loss.

3. RESULTS AND DISCUSSION

3.1. Results

In this section, we delve into the experimental results, commencing with an overview of the utilized dataset, shedding light on the intricacies of its composition and relevance to our study. Following that, we scrutinize the chosen evaluation metrics, providing insights into the quantitative measures employed to assess the model's performance. Subsequently, we present and analyse the obtained results, offering a comprehensive examination of the model's effectiveness in addressing the research objectives.

1. Dataset

In this section , we leverage the SQuAD (Stanford University Question Answering Dataset), a pivotal resource in natural language understanding and machine learning [12]. The evolved version, SQuAD 2.0, presents a more intricate challenge, featuring a training set with 130,319 instances and a development set with 11,873 instances. Through merging, our combined dataset comprises 142,192 instances. For our study, we focus on a sample data subset of 30,000 instances, emphasizing the essential attributes of questions and answer text. This selective dataset is crucial for addressing the specific requirements of our research, as depicted in Table 3.

Table 3. Sample from the used Dataset

No.	id	title	question	answer_ text	answer_ start	context
1	56be85543aeaaa14008c9063	Beyoncé	When did Beyoncé start becoming popular?	in the late 1990s	269	Beyoncé Giselle Knowles-Carter
2	56be85543aeaaa14008c9065	Beyoncé	What areas did Beyoncé compete in when she was growing up?	singing and dancing	207	Beyoncé Giselle Knowles-Carter

2. Evaluation materials

The system's performance is assessed using the following metrics, each accompanied by its respective equation, as presented below ([13], [14], [15], [16], [17]):

$$LOSS: Crossentropy = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} yij \log(pij) \quad (1)$$

$$\text{REC: Recall} = \frac{TP}{TP+TN} \quad (2)$$

$$\text{PREC: Precision} = \frac{TP}{TP+FP} \quad (3)$$

$$\text{F1: F1 Score} = \frac{2*presion*recall}{presion+recall} \quad (4)$$

$$\text{ACC: Accuracy} = \frac{TP+TN}{TP+TN+FN+FP} \quad (5)$$

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

$$EM\text{: Exact Match} = \frac{\text{Number of questions with exact match}}{\text{Total number of questions}} \times 100 \quad (6)$$

$$\text{MAP: Mean Average Precision} = \frac{1}{Q}\sum_{i=1}^{Q}\frac{1}{ranki} \quad (7)$$

$$\text{MAP: Mean Average Precision} = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{num\_coreccti}\sum_{j=1}^{num\_coreccti}\frac{j}{ranki} \quad (8)$$

where:

- N is the number of samples in the dataset.
- C is the number of classes.
- yi is the true value (1 if class j is the correct class for item i, and 0 otherwise).
- pij is the expected probability that item i is in class j according to the model.
- TP the number of true positive results
- TN the number of true negative results
- FP the number of false positive results
- FN the number of false negative results
- |Q| is the number of queries or items in the list.
- Q is the total number of queries.
- rranki is the rank position of the first correct answer for the i-th query.
- ranki is the rank of the correct item for query i.
- num_correcti is the number of correct items for query i.
- rankij is the rank of the j-th correct item for query i.
- Number of questions with exact match: The count of questions that were answered correctly in their entirety.
- Total number of questions: The total number of questions in the dataset.

3. *Eexperimental results*

Within this section, we elucidate the remaining stages outlined in the methodology, encompassing training, validation, evaluation, and prediction.

A. Training and Validation

During this phase, the training dataset, representing 70% of the selected data sample, underwent training, and the training process was assessed using the validation set, constituting 15%. These operations were conducted with specific parameters (150 epochs, batch size 128, embedding 100, and Lstm_units 256). Notably, favourable results were achieved in terms of accuracy and loss metrics. It is noteworthy that alternative configurations, such as varying epochs, batch sizes, embedding values, Lstm_units, and dataset partition ratios (80% training, 10% validation, and 10% testing), were explored. However, the optimal outcomes, in terms of accuracy, loss, and correct answer predictions, were consistently observed with the initially mentioned parameters. The loss measurement ratios are illustrated in Figure 6, and the accuracy measurement ratios in Figure 7 depict the variations in accuracy and loss measurements throughout each epoch (150). Let us now analyze these figures as follows:

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

i. Figure 6 for training and validation loss:

- Initial drop: There is a sharp drop in loss for both training and validation initially, which quickly stabilizes. This indicates that the model learns quickly from the raw data.
- Convergence: As epochs increase, the training and validation loss lines get closer, showing that the model is not overfitting and generalizes well to the validation set.
- Steady decline: The loss gradually decreases over epochs, and eventually stabilizes, indicating that the model may be reaching its optimal learning ability.

ii. Figure 7 for training and accuracy verification:

- Rapid increase: There is a sharp increase in accuracy initially, indicating rapid adaptation of the model to the features of the dataset.
- Continuous improvement: The accuracy of both training and validation is constantly improving, with a slight increase in training accuracy, which is typical for machine learning models.
- Stabilization trend: At the end of the training process the rate of increase in accuracy decreases and appears to reach a plateau, indicating that continued training may not lead to significant improvements.

In both figures, the fact that the validation metrics follow the training metrics closely indicates that the model is performing well rather than over-performing. Choosing epoch 150 as the cutoff point is reasonable since both loss and accuracy are constant, indicating that the model has likely learned as much as possible from the data provided. Any additional training will likely result in minimal gains and may risk overfitting the training data.

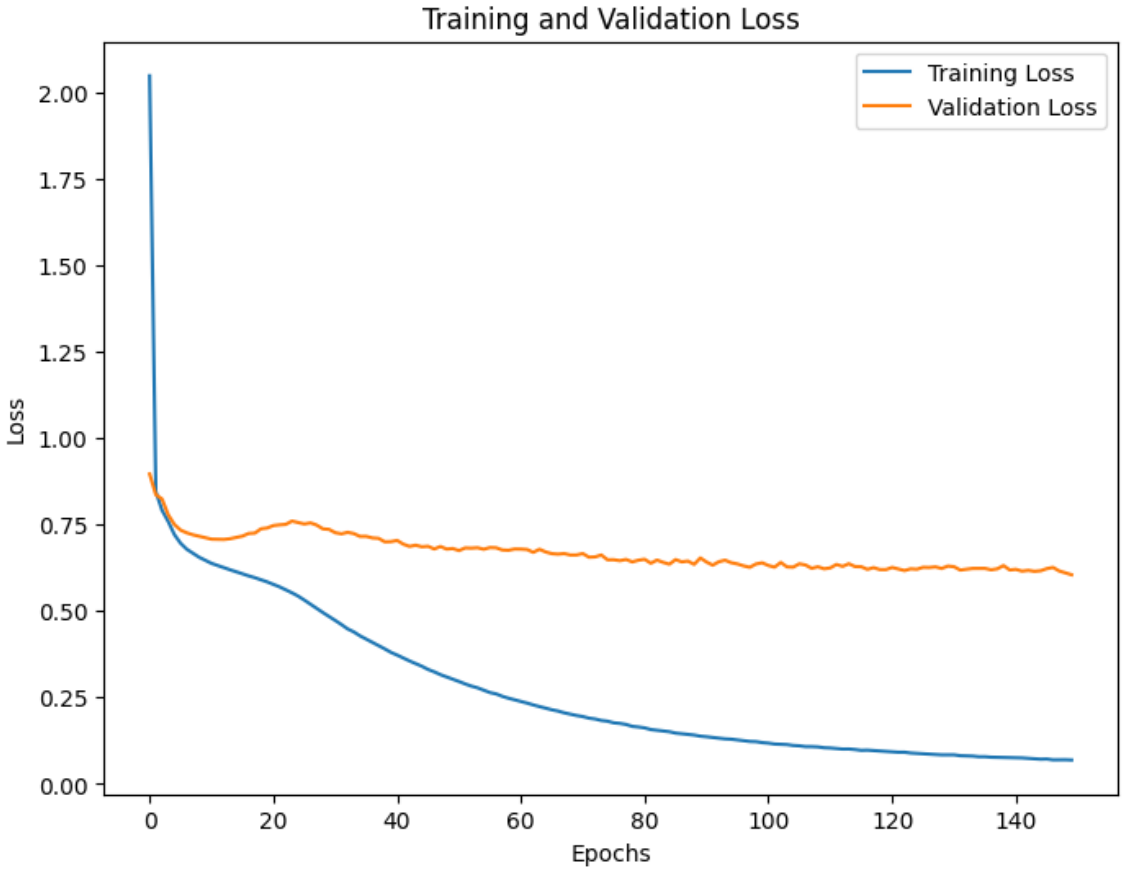

Fig. 6. Training and Validation Loss

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

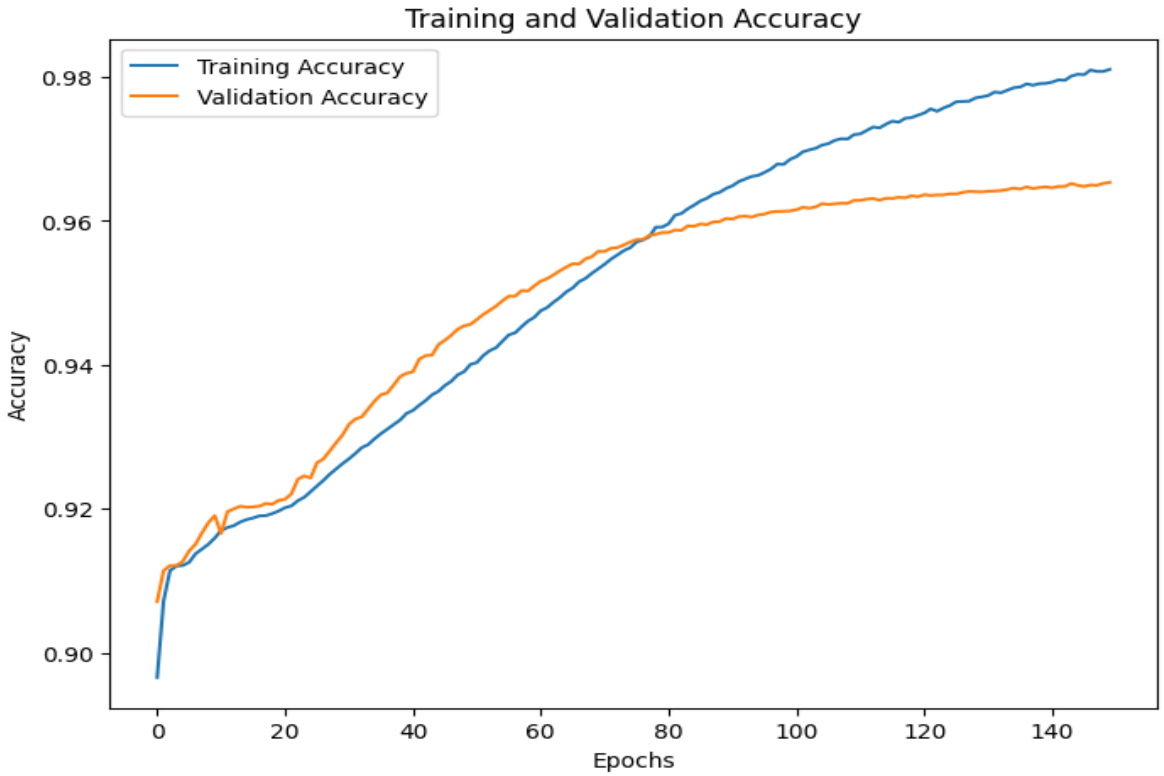

Fig. 7. Training and Validation Accuracy

B. Evaluation

During this part, the system's performance was rigorously assessed using the evaluation metrics detailed in Section 3.2. Exceptional results were obtained, particularly with the accuracy measure, reaching an impressive percentage of 96%. This percentage signifies a noteworthy achievement compared to other system metrics. The excellence of these metrics not only lies in numerical ratios but also in the system's proficiency in accurately predicting answers to posed questions. Detailed evaluation results are outlined in Table 4, while Figure 8 provides a comparative overview of the performance measures employed.

Table 4. The Results for Evaluation the Model by Metrics

No. Epochs	LOSS	ACC	PREC	REC	F1	EM	MRR	MAP
10	0.689	0.922	0.902	0.922	0.911	0.922	0.438	0.764
25	0.777	0.930	0.928	0.930	0.928	0.930	0.495	0.783
50	0.761	0.943	0.943	0.943	0.942	0.943	0.598	0.819
75	0.645	0.956	0.955	0.956	0.955	0.956	0.698	0.858
100	0.675	0.961	0.960	0.961	0.960	0.961	0.736	0.873
125	0.658	0.963	0.963	0.963	0.963	0.963	0.750	0.880
150	0.591	0.966	0.966	0.967	0.966	0.967	0.776	0.891
175	0.750	0.964	0.963	0.964	0.963	0.964	0.750	0.881
200	0.642	0.967	0.966	0.967	0.966	0.967	0.776	0.891

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

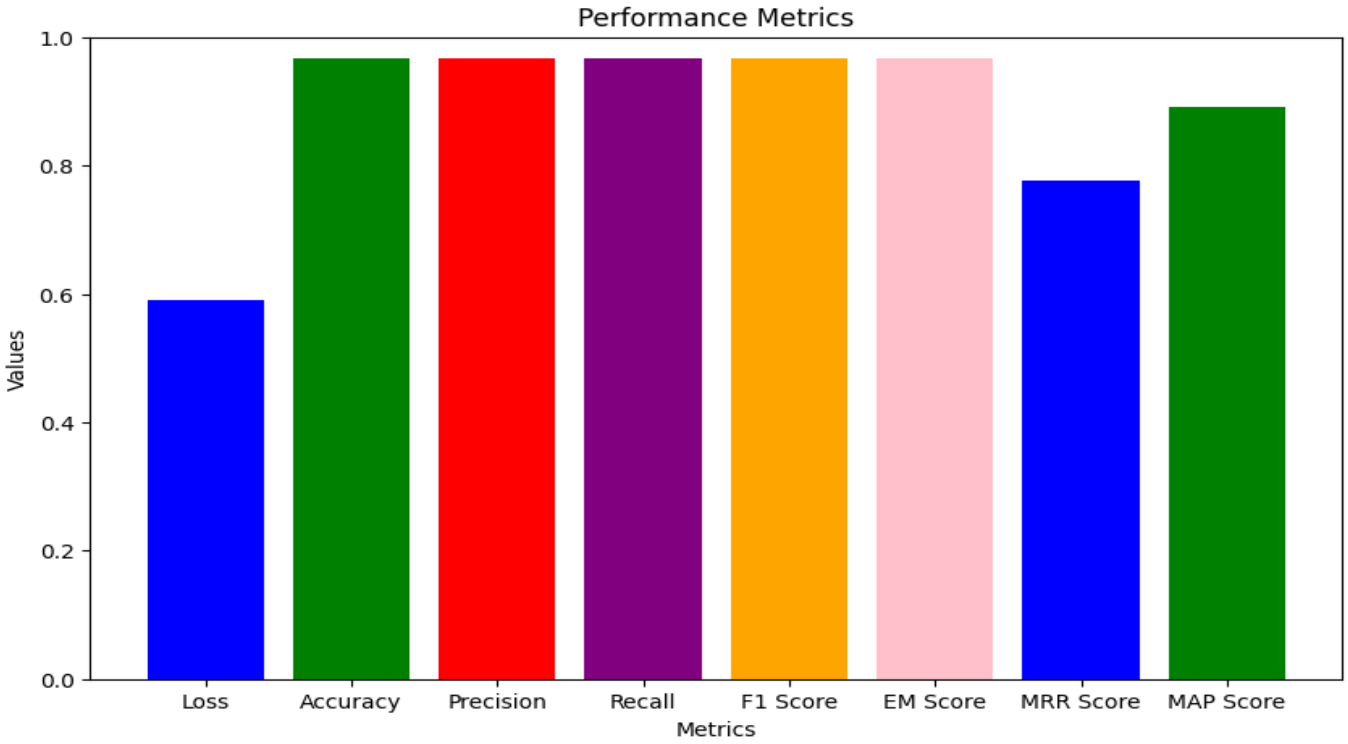

Fig. 8. The Results using Different Performance Metrics

Analysis of the model performance over the different eras as shown in Table 4 shows significant improvements in all metrics. Here is a brief summary:

- **Loss**: There is a steady decrease in loss from 0.689 at epoch 10 to 0.591 at epoch 150, indicating enhanced model learning and reduced prediction errors.
- **Accuracy**: Accuracy improves from 0.922 at epoch 10 to 0.967 at epoch 200, demonstrating the model's increasing ability to correctly predict answers.
- **Precision and Recall**: Both metrics improve in tandem, maintaining a good balance between minimizing false positives and false negatives.
- **F1 Score**: This metric, which combines precision and recall, shows consistent improvement, indicating effective simultaneous optimization of both metrics.
- **EM** and **MRR**: Both metrics show steady improvements, highlighting the model's proficiency in providing precise answers and ranking correct answers higher.
- **MAP**: Improvement in MAP suggests better ranking of provided answers, reflecting a deeper understanding of input texts.

Epoch 150 was chosen based on optimal performance ratios in accuracy and loss, as well as the model's good predictive ability, making it the most efficient point for model evaluation and deployment.

C. Prediction

In the last stage of building model for the question and answer system, a specific question is entered and the corresponding response is obtained. The model exhibited a notable proficiency in predicting correct answers, showcasing an average error rate for specific questions entered, as detailed in Table 5.

Table 5: sample of model predictions.

No.	Prediction	Prediction validity
1	Question: In which decade did Beyonce become famous?	True
	True Answer: late 1990s	
	Predicted Answer: late 1990s	
2	Question: To set the record for Grammys, how many did Beyonce win?	True
	True Answer: six	
	Predicted Answer: six	
3	Question: Which of her teachers discovered Beyonce's musical talent?	False
	True Answer: dance instructor Darlette Johnson	
	Predicted Answer: dance 236900 10000	

3.2. Discussion

After analysing the results obtained from this research, a comparison is drawn with another study [18] that used the same dataset but implemented the BERT technique. In our research, the BiLSTM technique was employed, showing superior performance compared to the comparative study. The F1 score achieved was 96.6%, and the Exact Match (EM) rate was 96.7%, significantly higher than the comparative study's F1 score of 77.816% and EM rate of 74.505%. This underscores the BiLSTM's enhanced ability to understand and respond to complex queries effectively, showcasing its superiority over the BERT technique in this context. While BERT is influential and widely applied within the field of question-and-answer systems, our findings highlight BiLSTM's greater efficacy for this particular dataset and application.

4. CONCLUSIONS

In conclusion, this research has effectively demonstrated the potential of Bidirectional Long Short-Term Memory (BiLSTM) in enhancing question and answer systems, achieving significant metrics such as 96.6% accuracy, 96.6% precision, 96.7% recall, 96.6% F1 score, and 96.7% Exact Match at epoch 150 with a batch size of 128 on the SQuAD 2.0 dataset. These results highlight the model's efficiency in text analysis and generating precise answers, reflecting BiLSTM's capability to handle linguistic complexities and contextual nuances. Moving forward, we plan to extend the application of our model to include multiple languages and integrate attention mechanisms to further enhance its ability to dynamically focus on the most relevant aspects of the input data. This will improve the model’s interpretability and effectiveness in complex scenarios. Additionally, we will explore the mathematical foundations of deep neural networks to stimulate more innovations in AI system architecture, which will enhance the practical utility of question answering technologies in real-world applications and help achieve our ongoing goal of minimizing losses while improving accuracy.

REFERENCES

[1] M. Li and Q. Wang, “*Question Answering on SQuAD2.0*”.
[2] Z. Huang *et al.*, “*Recent Trends in Deep Learning Based Open-Domain Textual Question Answering Systems*,” *IEEE Access*, vol. 8, pp. 94341–94356, 2020, doi: 10.1109/ACCESS.2020.2988903.
[3] X. Sun and X. Li, “*Research on Question Answering Technology Based on Bi- LSTM*,” *J. Phys. Conf. Ser.*, vol. 1325, no. 1, 2019, doi: 10.1088/1742-6596/1325/1/012009.
[4] B. Song, Y. Zhuo, and X. Li, “*Research on question-answering system based on deep learning*”, vol. 10942 LNCS. Springer International Publishing, 2018. doi: 10.1007/978-3-319-93818-9_50.

[5] Y. Zhang and Y. Peng, "*Research on Answer Selection Based on LSTM*," *Proc. 2018 Int. Conf. Asian Lang. Process. IALP 2018*, pp. 357–361, 2018, doi: 10.1109/IALP.2018.8629166.

[6] C. Li, L. Liu, and F. Jiang, "*Intelligent question answering model based on CN-BILSTM*," *ACM Int. Conf. Proceeding Ser.*, pp. 447–450, 2018, doi: 10.1145/3297156.3297261.

[7] C. B. GEMİRTER and D. GOULARAS, "*A Turkish Question Answering System Based on Deep Learning Neural Networks*," *J. Intell. Syst. Theory Appl.*, vol. 4, no. 2, pp. 65–75, 2021, doi: 10.38016/jista.815823.

[8] J. Yin, "*Research on Question Answering System Based on BERT Model*," 2022 3rd Int. Conf. Comput. Vision, Image Deep Learn. Int. Conf. Comput. Eng. Appl. CVIDL ICCEA 2022, pp. 68–71, 2022, doi: 10.1109/CVIDLICCEA56201.2022.9824408.

[9] B. Antony and N. R. Paul, "*Question Answering System for Tamil Using Deep Learning*," *Commun. Comput. Inf. Sci.*, vol. 1802 CCIS, no. 1001, pp. 244–252, 2023, doi: 10.1007/978-3-031-33231-9_17.

[10] H. Tian, H. Fan, M. Feng, R. Cao, and D. Li, "*Fault Diagnosis of Rolling Bearing Based on HPSO Algorithm Optimized CNN-LSTM Neural Network*," *Sensors*, vol. 23, no. 14, 2023, doi: 10.3390/s23146508.

[11] Z. Zhu, Q. Yang, X. Liu, and D. Gao, "*Attention-based CNN-BiLSTM for SOH and RUL estimation of lithium-ion batteries*," *J. Algorithms Comput. Technol.*, vol. 16, 2022, doi: 10.1177/17483026221130598.

[12] "*Dataset download*", [Online]. Available: https://rajpurkar.github.io/SQuAD-explorer/

[13] Z. Zhou, H. Huang, and B. Fang, "*Application of Weighted Cross-Entropy Loss Function in Intrusion Detection*," pp. 1–21, 2021, doi: 10.4236/jcc.2021.911001.

[14] T. Salman, A. Ghubaish, D. Unal, and R. Jain, "*Safety Score as an Evaluation Metric for Machine Learning Models of Security Applications*," vol. 2, no. 4, pp. 207–211, 2020, doi: 10.1109/LNET.2020.3016583.

[15] H. Bahak, F. Taheri, Z. Zojaji, and A. Kazemi, "*Evaluating ChatGPT as a Question Answering System: A Comprehensive Analysis and Comparison with Existing Models*," 2023, [Online]. Available: https://arxiv.org/abs/2312.07592v1

[16] H. Yoganarasimhan, "*Search personalization using machine learning*," *Manage. Sci.*, vol. 66, no. 3, pp. 1045–1070, 2020, doi: 10.1287/mnsc.2018.3255.

[17] R. Kusumaningrum, A. F. Hanifah, K. Khadijah, S. N. Endah, and P. S. Sasongko, "*Long Short-Term Memory for Non-Factoid Answer Selection in Indonesian Question Answering System for Health Information*," *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 2, pp. 381–388, 2023, doi: 10.14569/IJACSA.2023.0140246.

[18] Y. Li and Y. Zhang, "*Question Answering on SQuAD 2.0 Dataset*," *Dep. Stat.*, pp. 1–7, 2019, [Online]. Available: https://rajpurkar.github.io/SQuAD-explorer/