International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Intelligent Question Analyzer System: Embedding-based Model for Semantic Analysis

Alby Benny[a]*, Alwin P Jose*[a]*, Mikhela Theres Sabu Mathew*[a]*, Navaneeth V Nair*[a]*, Dr.Anita Brigit Mathew*[b]

[a] UG Student, Artificial Intelligence and Data Science, Viswajyothi College of Engineering and Technology, Muvattupuzha, 686661, India
[b] Head of Department, Artificial Intelligence and Data Science, Viswajyothi College of Engineering and Technology, Muvattupuzha, 686661, India

ABSTRACT:

Automating the correction process of answer sheets is pivotal in educational settings, particularly when questions involve the interpretation of figures. Accurate analysis of questions is crucial for proper evaluation and mark distribution. This paper proposes an Intelligent Question Analyzer System using TensorFlow, aimed at automating the identification of whether a question requires figures for evaluation. The model employs natural language processing (NLP) techniques for preprocessing questions, utilizing tokenization to convert text into tokens. By capturing semantic meaning and context, the model identifies patterns indicative of figure necessity. In the introduction, we outline the complexity of correcting answer sheets, especially when figures are involved. Traditional methods are inadequate for such tasks, necessitating neural network approaches for scalability and adaptability. Leveraging the processing power of current hardware, neural networks can effectively handle large datasets and complex tasks. The model can be fine-tuned for specific domains, mitigating the challenge of dataset preparation by employing techniques like data augmentation. Preprocessing involves converting questions to tokens and padding sequences to ensure uniform length. Neural networks excel in capturing intricate text relationships, enhancing prediction accuracy. By integrating network methods, this system aims to enhance the precision, efficiency, and fairness of educational evaluations, ultimately reducing teacher workload and fostering an improved learning environment.

Keywords: Intelligent Question Analyzer (IQA), Natural Language Processing (NLP), Support Vector Machine (SVM), K-Nearest Neighbors (K-NN)

1. Introduction

In the field of education, using AI to correct student answer sheets is a complicated step. But when questions involve problems or drawing figures, the regular correction process becomes more complex as the figure comparison process has to be done to properly evaluate the answers. To address this challenge, it is essential to first analyze each question to determine whether it require the need of a figure to be drawn. This initial analysis serves as a crucial precursor to the subsequent correction process, laying the groundwork for precise evaluation and mark distribution. In this context, our approach adopts a neural network methodology for question analysis, leveraging its inherent scalability and adaptability to effectively handle diverse datasets and complex tasks. Neural networks offer the advantage of being able to scale seamlessly to accommodate large datasets, tapping into the processing capabilities of contemporary computer hardware. Furthermore, they can be fine-tuned and tailored to specific educational domains, enhancing their applicability and performance.

However, the successful implementation of a neural network model for question analysis hinges on the availability of a comprehensive dataset and meticulous preprocessing of questions. While our dataset may be comparatively smaller due to focusing on a single subject, we mitigate this limitation through data augmentation techniques. Preprocessing involves tokenizing questions and ensuring uniform sequence lengths through padding, facilitating optimal model performance. Beyond scalability and adaptability, neural networks excel in capturing intricate relationships and patterns within textual data, leading to more accurate predictions regarding figure requirements. By amalgamating neural network methodologies, our objective is to enhance the precision, efficiency, and fairness of educational evaluations while alleviating the burden on educators and fostering an enriched learning environment for students.

2. Literature Survey

2.1 A Study of the Optimization Algorithms in Deep Learning

The study by Zaheer and Shaziya [1] delves into the optimization algorithms used in deep learning, focusing on The objective during the learning process is to minimize loss using stochastic gradient descent, along with techniques such as Nesterov momentum, RMSprop, Adam, Adagrad, and Adadelta by iteratively adjusting model parameters. The work discusses the importance of optimization algorithms in machine learning and deep learning, emphasizing their role in parameter learning and model accuracy improvement. Various studies have explored different optimization techniques aimed at tackling the difficulties inherent in the learning process. The paper examines datasets such as cifar10, cifar100, mnist, and fashion mnist to evaluate the performance of selected optimization algorithms. Results obtained from training models on these datasets are compared, providing insights into the effectiveness of different optimization techniques.

2.2 Question to Question Similarity Analysis Using Morphological, Syntactic, Semantic, and Lexical Features

This study [2] provide a foundation for the current research's focus on Conducting a similarity analysis between questions. using a combination of syntactic, morphological, lexical, and semantic features in the Arabic language. Previous research by Al-Anzi and AbuZeina highlighted the effectiveness of cosine similarity as a measure for Arabic text classification. This supports the decision to include cosine similarity as a Lexical characteristics in the model. AL-Smadi et al. introduced a method for detecting similar news within Arabic Tweets on Twitter, showcasing the relevance of similarity detection in Arabic text. Hamza et al. developed a categorization system for Arabic question areas and a classification technique to aid question answering platforms in efficiently retrieving answers. They introduced an approach that leveraged semantic knowledge of words for text classification, using supervised algorithms like K-Nearest Neighbors (K-NN) and Support Vector Machine (SVM). While previous studies have explored various aspects of Arabic text classification, the specific problem addressed in the current research - determining the similarity between two Arabic questions efficiently and accurately - remains a research challenge.

3. Proposed work

3.1 Intelligent Question Analyzer

Here we introduce a neural network model for analyzing the question and predicting whether a figure is required or not. The Intelligent Question Analyzer (IQA) system uses natural language processing (NLP) techniques to prepare the questions before analyzing them with networks ensuring that the data input is correctly structured. We are going to build a Intelligent Question Analyzer (IQA) system which is trained on a custom dataset. By using datasets designed for particular educational fields IQA seeks to offer precise and dependable forecasts on the necessity of the figures in question thereby simplifying and enhancing assessment procedures. Given the relatively small model complexity, the training of IQA can be conducted on conventional CPUs, making it accessible and feasible for educational institutions with limited computational resources.

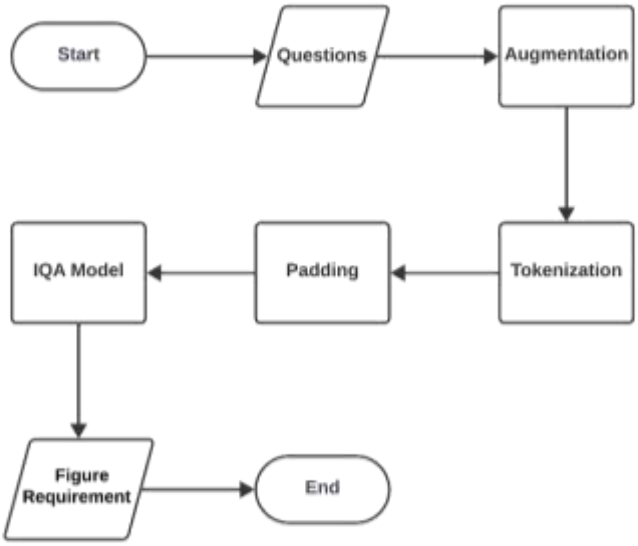


Fig. 1 Overview of overall Process

3.2 Model Overview

As shown in the figure 2, the model architecture consist of 7 layers: Embedding Layer, Pooling Layer, Dense Layer 1, Dropout Layer 1, Dense Layer 2, Dropout Layer 2, Output Layer.

1) Embedding Layer: It converts integer-encoded sequences (representing the words) into fixed-size dense vectors.

2) Pooling Layer: We use the GlobalAveragePooling1D as the pooling layer. The GlobalAveragePooling1D layer averages over the time dimension, effectively reducing the sequence length to 1. This operation aggregates information from all time steps of the sequence into a single vector.

3) Dense Layer 1: This dense layer consists of 64 neurons and the ReLU (Rectified Linear Unit) activation function introduces non-linearity to the model.

4) Dropout Layer 1: During training, dropout, a regularization technique, randomly sets a portion of input units to zero. Its purpose is to mitigate overfitting by diminishing the co-adaptation among neurons. Here we use the dropout layer with 50% dropout rate.

5) Dense Layer 2: This dense layer consists of 32 neurons and ReLU (Rectified Linear Unit) activation function is again used.

6) Dropout Layer 2: Another dropout layer with a dropout rate of 50%.

7) Output Layer: The output layer is a single neuron equipped with a sigmoid activation function ensures that the output is flattened between 0 and 1, representing the probability that the question requires an image.

The figure 3. Shows the model architecture and the shape of each layer with their parameters.

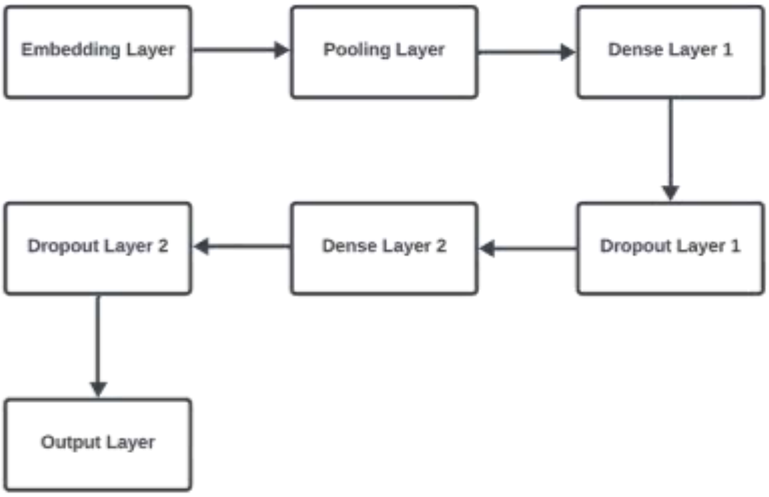
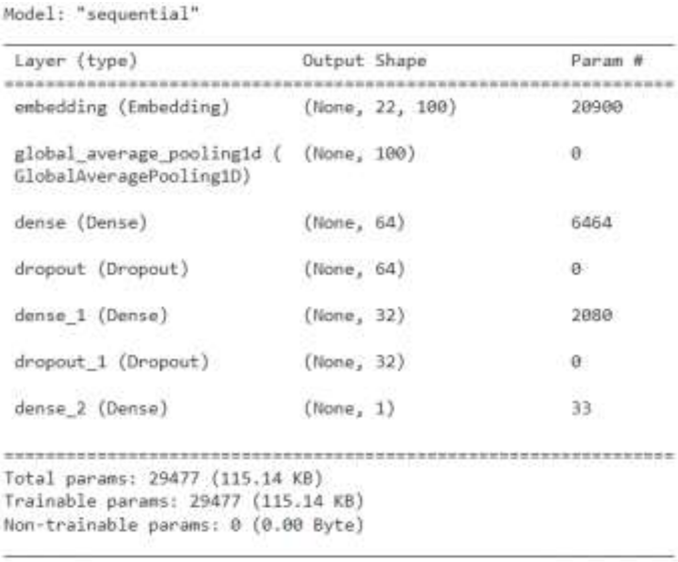


Fig. 2 Overview of IQA Model



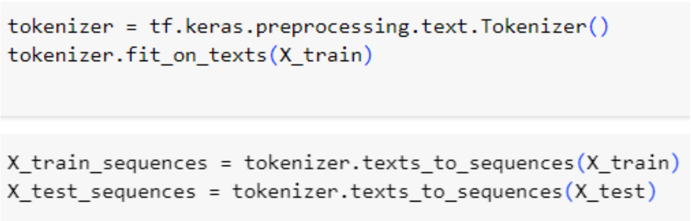
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 22, 100)           20900

 global_average_pooling1d (  (None, 100)               0
 GlobalAveragePooling1D)

 dense (Dense)               (None, 64)                6464

 dropout (Dropout)           (None, 64)                0

 dense_1 (Dense)             (None, 32)                2080

 dropout_1 (Dropout)         (None, 32)                0

 dense_2 (Dense)             (None, 1)                 33

=================================================================
Total params: 29477 (115.14 KB)
Trainable params: 29477 (115.14 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Fig. 3 Model Architecture

3.3 Tokenization

Tokenization involves converting text data into a numerical data that can be understood by ML models. Here we use basic word-level tokenization technique to convert the text data to numerical data. This implies that every word in the sentence is treated as an individual token, with the tokenizer assigning a distinct index to each word within the vocabulary.

As shown in figure 4. first we initialize a tokenizer object using the tf.keras.preprocessing.text.Tokenizer() method. Then we fit the tokenizer on the training text data using fit_on_texts() function. The texts_to_sequences() method converts the text data into sequences of numbers. Each word in the text is replaced with its corresponding index in the vocabulary, creating sequences of indices representing the words in the text.



```
tokenizer = tf.keras.preprocessing.text.Tokenizer()
tokenizer.fit_on_texts(X_train)
```

```
X_train_sequences = tokenizer.texts_to_sequences(X_train)
X_test_sequences = tokenizer.texts_to_sequences(X_test)
```

Fig. 4 Tokenization Step

3.4 Padding

Padding is a preprocessing step commonly used in natural language processing (NLP) tasks, particularly when working with sequences of variable lengths. In the case of tokenized data, padding ensures that all sequences have the same length, which is necessary for neural network models that expect fixed-size inputs. The tokenized input data from the tokenizer may vary in length. When a sequence is shorter than the longest sequence in the training set, zeros are added either at the beginning or end of the sequence to match the length of the other sequences. This is done using the pad_sequences() method which takes the sequence and maxlen as input and gives the padded sequence as output.

3.5 Model compilation and Training

Here we compile the proposed model with the adam optimizer, binary crossentropy loss and the accuray metrics. The optimizer determines how the model's parameters (weights and biases) are updated during training to minimize the loss function. Adam is a popular optimization algorithm that adapts the learning rate during training based on the exponentially decaying average of past gradients. It combines the advantages of both AdaGrad and RMSProp optimizers.

During training, the loss function measures the degree to which the model's predictions align with the real target labels. It measures the discrepancy between predicted output and ground truth. For binary classification tasks, the loss function commonly utilized is the binary cross-entropy. Binary cross-entropy loss is specifically designed for binary classification problems. It computes the cross-entropy loss between the true labels and the predicted probabilities, penalizing large deviations between the predicted and actual class probabilities.

Metrics are used to evaluate the model’s performance during training and testing. Here we use, accuracy as the metric. Accuracy quantifies the ratio of correctly classified sample datas relative to the total number of samples. During training, the accuracy metric will be computed and displayed for each epoch, helping to monitor the model's performance and convergence.

For the training process, we will use 50 epochs, a batch size of 32 and a vaildation dataset. During the training process, an epoch denotes a full iteration through the entire training dataset. In each epoch, the model goes through all the training examples, computes the loss, and the optimization algorithm (adam) updates the model's parameters, while the batch size represents the number of training examples utilized in a single iteration of the training process.

4. Results and Discussion

Figure 5. shows the plot of training loss and the validation loss against epochs. The convergence of both lines as epochs increase suggests that the model is not overfitting and is improving its performance. Overfitting happens when a model demonstrates strong performance on the training data but struggles when presented with unseen data. The plateauing of the training loss also indicates that the model might not benefit from further training on the current dataset or might require some adjustments to improve further.

Similarly the figure 6. shows the training accuracy and the validation accuracy against epochs. After completing 50 epochs, the model have accuracy around 97% and validation accuracy around 92%. After analyzing the plots, we can understand that there no point in increasing the epochs as it may lead to the overfitting condition. When evaluated with the test dataset, we got the overall accuracy of the model around 95%, which makes it prominent for predicting the output for the specific subject.

5. Conclusion and Future Scope

In this study, we develop a model for analyzing the question for predicting the figure requirment. Predicting the figure requirment is a vital step for evaluating hand drawn figure. The IQA system is developed using neural network hence the system is flexible and can be adapted to various domains and tasks. The versatility of the IQA system renders it appropriate for various applications, including educational assessments, image recognition tasks, and other domains where question analysis is essential. Furthermore, the use of neural networks enables the system to continuously improve its performance through techniques such as fine-tuning and retraining on new data, ensuring its relevance and efficacy over time. Overall, our study presents a robust and versatile solution for question analysis, paving the way for enhanced automation and efficiency in various fields.

The future scope for the Intelligent Question Analyzer (IQA) system is vast and promising.

1) Enhanced Question Analysis: One avenue for future development involves expanding IQA's capabilities to classify questions beyond their need for figures. This entails predicting whether a question is problematic, requiring critical thinking or problem-solving skills, or theoretical, focusing on conceptual understanding or knowledge recall.

2) Generalization to Other Languages: Extending the IQA system to support analysis of questions in languages other than English. This would require training the model on multilingual datasets and adapting it to handle the linguistic characteristics of different languages.

3) Integration with Educational Platforms: Integrating the IQA system with existing educational platforms or learning management systems (LMS) to seamlessly incorporate question analysis capabilities into educational workflows. This could streamline the assessment process and provide valuable insights to educators and students.

4) Multimodal Analysis: Integrating multiple modalities such as text, images, and possibly audio to provide a more comprehensive analysis of questions. This could involve extending the model architecture to handle multimodal inputs and predicting the necessity of figures based on both textual and visual information.

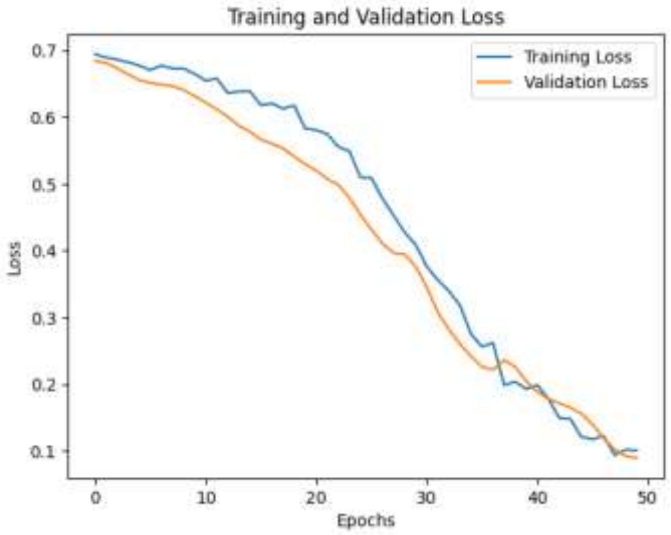


Fig. 5 Plot of Loss against Epochs

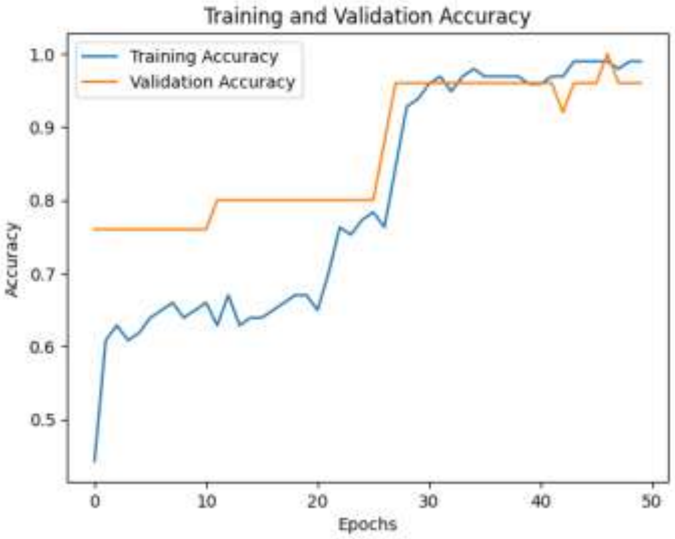


Fig. 6 Plot of Accuracy against Epochs

Acknowledgment

We extend our sincere gratitude to Viswajyothi College of Engineering and Technology, Vazhakulam, for their invaluable assistance and unwavering support throughout this endeavor. Our heartfelt appreciation goes to our esteemed Project Coordinators, Mrs. Mary Nirmala George and Mrs. Anupriya Mohan, for their continuous motivation, guidance, and support. Additionally, we extend our thanks to all the faculty members of the Artificial Intelligence and Data Science Department for their enduring encouragement and assistance.

REFERENCES

[1] Zaheer, R., & Shaziya, H. (2019, January). A study of the optimization algorithms in deep learning. In 2019 third international conference on inventive systems and control (ICISC) (pp. 536-539). IEEE.

[2] Al-Asa'd, M., Al-Khdour, N., Younes, M. B., Khwaileh, E., Hammad, M., & Mohammad, A. S. (2019, November). Question to Question Similarity Analysis using Morphological, Syntactic, Semantic, and Lexical Features. In 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA) (pp. 1-6). IEEE.

[3] Nalawade, G., & Ramesh, R. (2016, December). Automatic generation of question paper from user entered specifications using a semantically tagged question repository. In 2016 IEEE Eighth International Conference on Technology for Education (T4E) (pp. 148-151). IEEE.

[4] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. Advances in neural information processing systems, 27.

[5] Pisat, P., Modi, D., Rewagad, S., Sawant, G., & Chaturvedi, D. (2017, August). Question paper generator and answer verifier. In 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS) (pp. 1074-1077). IEEE.

[6] Mielke, S. J., Alyafeai, Z., Salesky, E., Raffel, C., Dey, M., Gallé, M., ... & Tan, S. (2021). Between words and characters: A brief history of open-vocabulary modeling and tokenization in NLP. arXiv preprint arXiv:2112.10508.

[7] Hrinchuk, O., Khrulkov, V., Mirvakhabova, L., Orlova, E., & Oseledets, I. (2019). Tensorized embedding layers for efficient model compression. arXiv preprint arXiv:1901.10787.