Can Multi-Modal models help pass your science class?

Naigam Shah
Department of Computer Science and Engineering
University of California San Diego
n7shah@ucsd.edu

Adithya Samavedhi
Department of Computer Science and Engineering
University of California San Diego
asamavedhi@ucsd.edu

Ganeshan Malhotra
Department of Electrical and Computer Engineering
University of California San Diego
gmalhotra@ucsd.edu

Mayank Sharma
Department of Electrical and Computer Engineering
University of California San Diego
musharma@ucsd.edu

Shreya Sumbetla
Department of Computer Science and Engineering
University of California San Diego
ssumbetla@ucsd.edu

Aishwarya Manjunath
Department of Computer Science and Engineering
University of California San Diego
aimanjunath@ucsd.edu

Abstract

Humans often combine information from different modalities to form a coherent argument to answer a question. In this work we study this problem of question answering using the ScienceQA dataset which has text based hints and images as extra contextual features. We explore several methodologies including Siamese RNNs, Siamese LSTMs, pretrained Transformer based classification models, and pretrained Transformer models trained using vanilla contrastive loss and margin ranking loss functions. The siamese RNN and siamese LSTM models act as baselines achieving test accuracies of 34.87% and 52.08% respectively. The BERT and RoBERTa based classification models show a considerable imporvement over the RNN models achieving test accuracies of 72.01% and $78/02\%$ respectively. These models also benefit from text based hints in the dataset which helps the model to understand extra information which may be helpful to answer the questions. Our

experiments show that the Transformer models trained using margin ranking loss function are the best performing, achieving a test accuracy of 80.48% with the DistilRoBERTa model when trained with textual hints as extra contextual features, which suggests that the use of contrastive based loss functions can improve the performance of QA models, particularly when using pretrained Transformer-based architectures.

1 Introduction

The task of visual question answering is an important task which requires information from both visual and textual modalities. This is a challenging task since it involves answering open-ended questions that require consolidating information from multiple modalities. But the advent of attention based methods (intra-modality attention and inter-modality based methods) [21] and transformer based methods [9], [18], has greatly improved the performance of models on this task. Recently, a lot of work has been done to model this task using Graph Neural Networks [23]. Some works also generate conterfactual examples to generate answers [3]. In this study, we want to focus on this task of Visual Question Answering. We will create a combined vision and language based model using state of the art pretrained language models. We want to study if these pretrained language models can generalize to scientific domain knowledge through our use of the Science-QA dataset [11]. We will also attempt to study the interpretability of these models when they answer the questions thereby trying to understand the rationale behind the answers given by these models. Lastly, we will also attempt to study the robustness of these models by providing adverserial examples to our model where the options in the answers do not match the description in the question and the visual information presented alongside with it.

2 Related works

Question Answering (QA) models have been a topic of extensive research in natural language processing (NLP). The aim of QA models is to provide human-like answers to questions posed in natural language. Different types of QA models include extractive QA, abstractive QA, and generative QA. The Watson system, proposed by IBM, was one of the earliest QA models that utilized various techniques such as information retrieval, natural language processing, and machine learning to answer questions (Ferrucci et al., 2010) [7].

In recent years, deep learning techniques have been extensively used in QA models. For example, Long Short-Term Memory (LSTM) networks, Convolutional Neural Networks (CNNs), and Attention Mechanisms have been used to develop QA models (Seo et al., 2017 [17]). Commonly used datasets for QA include SQuAD (Rajpurkar et al., 2016) [14], TriviaQA (Joshi et al., 2017) [8], MS MARCO (Nguyen et al., 2016) [12], and CoQA (Reddy et al., 2019) [16]

T5-QA is a recent QA model proposed by T5 language model fine-tuned on SQuAD dataset (Zhang et al., 2020) [22] The model achieved state-of-the-art performance on the SQuAD dataset. Similarly, BERT-Base is another QA model that utilizes BERT language model fine-tuned on SQuAD dataset (Devlin et al., 2018) [5] The BERT-Base model has also achieved state-of-the-art performance on the SQuAD dataset.

In conclusion, QA models have been extensively studied in the literature, and various models have been proposed utilizing different deep learning techniques. SQuAD, TriviaQA, MS MARCO, and CoQA are some of the commonly used datasets for QA. Recent studies have proposed T5-QA and BERT-Base models that have achieved state-of-the-art performance on the SQuAD dataset.

3 Dataset

The Science QA dataset [11] is a large-scale question answering dataset that focuses on answering questions related to scientific topics. It was created by the Allen Institute for AI (AI2) and contains over 15,000 questions and their corresponding answers, covering a wide range of scientific topics such as biology, chemistry, physics, and astronomy.

The dataset is designed for various tasks such as open-domain question answering, multi-hop reasoning, and knowledge extraction. One task that can be performed on the dataset is machine comprehension, where the system is trained to understand the meaning of the question and the relevant information in the given passage to generate an accurate answer. Another task is knowledge extraction, where the system is trained to extract relevant information from the passage and use it to answer the question. Additionally, the dataset can also be used for question generation, where the system generates questions based on the given passage.

The Science QA dataset is unique in that it focuses on scientific topics, which require a more specialized knowledge base and domain-specific reasoning abilities. This makes it a challenging dataset for natural language processing (NLP) systems, and it has been used as a benchmark for evaluating the performance of various NLP models.

3.1 Data Preprocessing and Analysis

We conducted exploratory data analysis of the Science QA dataset. The dataset includes multiple types of questions including MCQs (with varying number of options), yes/no questions and true or false questions. We have chosen to work with only MCQ data as they form the majority of the dataset. We extracted the MCQ data(with attribute task=closed_choice) and discarded the rest. We have presented the filtered data in this section. Some of the examples also contain images as part of the question (9984 with images and 10420 without images) (displayed in Figure 1). An example of a data entry is given in the following image (Figure 1).

Question	Which type of force from the baby's hand opens the cabinet door ?
Choices	A) Push B) Pull
Hint	A baby wants to know what is inside of a cabinet. Her hand applies a force to the door and the door opens
Answer	B) Pull

Question	Which of these states is farthest north ?
Choices	A) Oklahoma B) Arizona C) Louisiana D) West Virginia
Hint	-
Answer	D) West Virginia

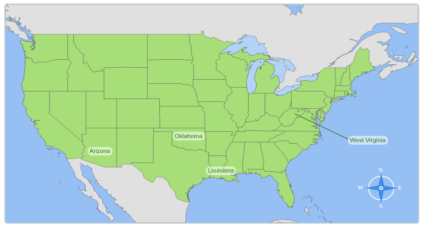

Figure 1: Sample dataset entries

The dataset includes different levels of difficulty of questions organised into grades 1-10. The distribution of the same is given in Figure 2. Majority of the data contains Grade 2-6 related questions. Each example is also tagged with the unique category that it belongs to including States, Basic economic principles, Particle motion and energy etc.

We analysed the distribution of answer choices in the dataset to investigate if the answer choices are skewed to any particular index. The distribution is given in the pie chart in Figure 2. The pie chart in Figure 2 indicates that for the majority of questions, the answer options are either A or B. However, this is not due to any bias towards these options. In fact, the reason for this distribution is that each question has a varying number of choices, with an average of 2.42 options per question. As a result, the majority of questions have just two choices. It is worth noting that for our specific use-case, the order of options does not matter since we are only interested in comparing the similarity of options

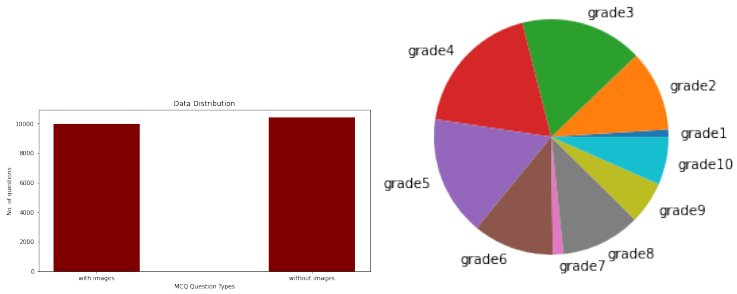

Figure 2: (a) Distribution of Data samples with and without images (b) Grade distribution of Science QA dataset

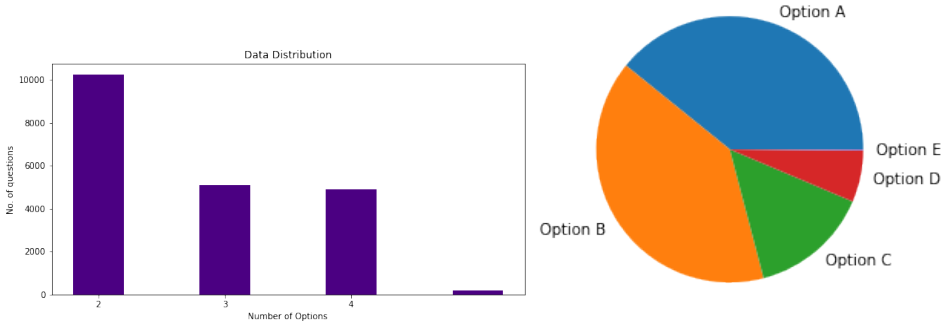

Figure 3: (a) Distribution of number of choices (b) Distribution of correct answer choices

Task	Train	Validation	Test
With images	6031	2004	1949
Without images	6223	2056	2141

Table 1: Train - Val - Test Distribution

Attribute	Description	Non-null count	Datatype
question	text of the question	20404	String
image	Image(part of the question) if present	9984	PIL Image
choices	List of multiple choice options	20404	List of string
answer	index of correct option	20404	Integer
hint	supporting text to help solve the question	9872	String
task	question type ex: closed-choice,T/F	20404	String(category)
grade	Grade 1-10	20404	String(category)
subject	school subject ex. social/natural sciences	20404	String(category)
topic	topic within subject	20404	String(category)
category	subtopic	20404	String(category)
skill	required skill for question	20404	String
lecture	Context for answering question	17029	String
solution	Instructions to solve question	18398	String

Table 2: Dataset Attributes

with given input embeddings. Therefore, even if we were to shuffle the options, the resulting output would remain the same.

4 Methodology

In this section, we discuss the methodologies used to help answer questions from ScienceQA. Firstly, we explore a LSTM based architecture using a siamese network as our baseline. We then explore to exploit the advances from transformer models by viewing the MCQA problem from the eyes of a classification problem. We then modify our approach to incorporate a cosine embedding loss and use cosine similarities for prediction.

4.1 Baseline Model: Siamese Network for Text Similarity

Siamese network as mentioned in [4], are neural networks which consists of two networks which share the same weights. The model is trained to ensure that inputs from similar categories are closer together and inputs from dissimilar categories are farther from each other. These networks can be very useful to learn the semantic similarity between two texts and can be applied to text or vision domains in order to measure similarity in text or visual data.

In this project, we are leveraging Siamese networks in order to measure the similarity score between question and the answer. In the context of question-answering - the question and the correct answer pair should have high similarity and the question and wrong answer pair should have low similarity score. The two input texts to the model (question + correct/incorrect answer) into fixed-length feature vectors. The feature vectors are then compared using a similarity metric, such as cosine similarity, to compute the similarity score of the pair.

In our initial implementation,we have encoded the input texts using a word embedding model - GloVe [13], which maps each word to a high-dimensional vector representation. Each of the question and answer texts are padded to be of length 30. This vector representation of each of the texts are fed as inputs to the Siamese network which is trained to learn the similarity score between the two input texts.

In order to understand how good the text information provided in the questions are without any dependency on images, we have implemented a Siamese model by considering the question and answer choices only. We have incorporated a model with RNN units and a model with LSTM units to capture temporal semantic similarity in the text. The model is shown below in Figure 1.

Figure 5 and Figure 4 shows the architecture of the Siamese network. As seen from the architecture, the input text embedding is passed to an RNN/LSTM layer which is forwarded to a dense fully connected linear layer. Contrastive loss between the question and answer output embedding is used during backpropogation. This ensures that the question and correct answer pair have high similarity and the question and wrong answer pair has low similarity. Contrastive loss is defined as below:

$$L = (1 - Y) * \frac{1}{2} * (D_w)^2 + (Y) * \frac{1}{2} * max(0, m - D_w)^2 \quad (1)$$

where m - is the margin and is set to 1, D_w is the Euclidean distance between the embeddings of the output of Siamese network. Y is set to 1 for the question and correct answer pair and is set to 0 for question and wrong answer pair.

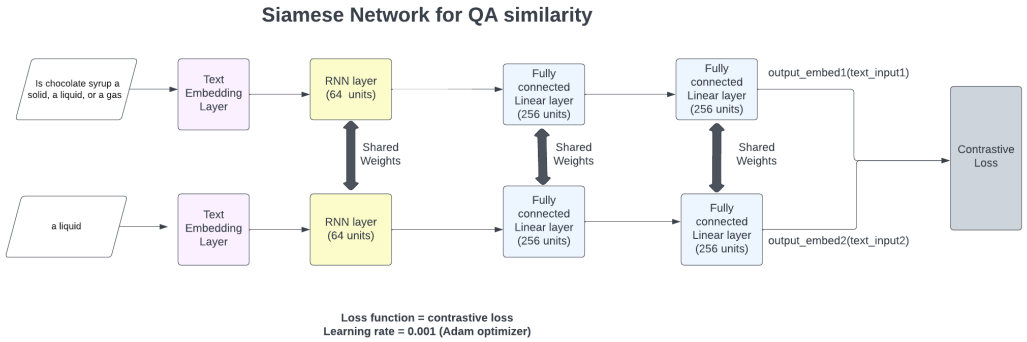

Figure 4: Baseline: Siamese Network architecture with RNN units

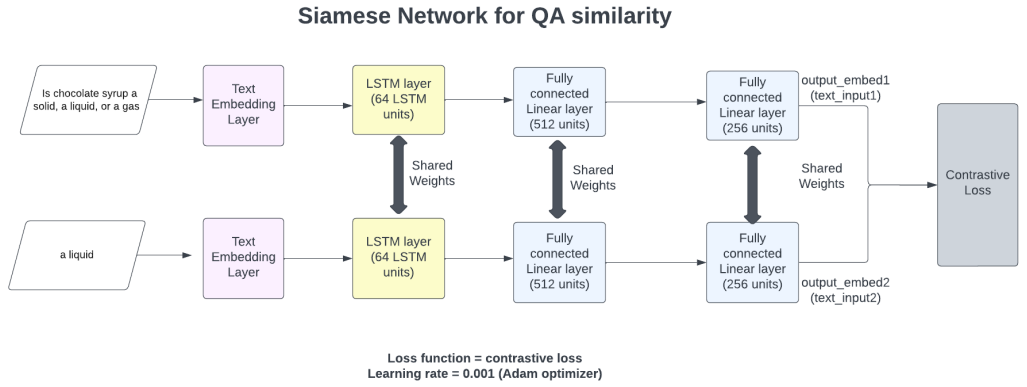

Figure 5: Baseline: Siamese Network architecture with LSTM units

4.2 Refactorization of Multi Choice Question Answering to Binary Question Answering

Transformer architecture were introduced in [19] and have revolutionized many Natural Language Processing tasks. Transformers are based on the encoder-decoder based architecture as shown in 6 and use attention mechanism to map queries to keys and values. BERT [6] is an autoencoder based

model based on the original transformer architecture. It was pretrained using the masked language modelling task.

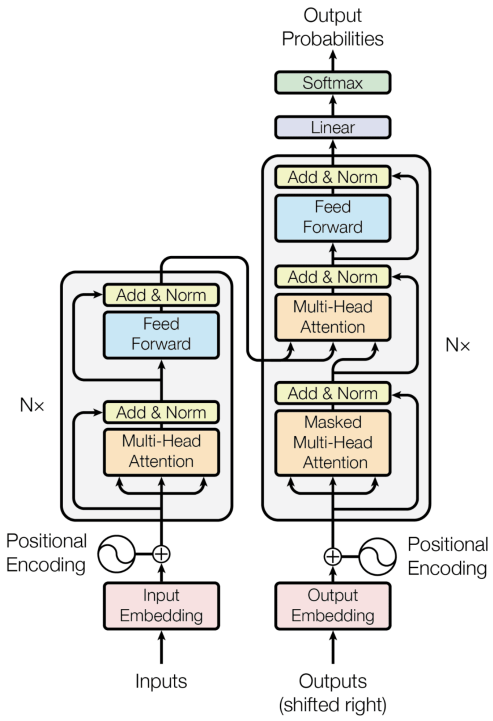

Figure 6: The original Transformer architecture

In this formulation, we refactor this task into a binary choice question answering task by exploding the data where we pass the question and each option to the model along with binary labels. This label is 1 if the choice is the correct choice and 0 if the choice is the wrong one.
We are using a pre-trained RoBERTa architecture using the huggingface library [1] with added linear layer at the top to act as the classifier.

4.2.1 RoBERTa base NLI

RoBERTa [10] is a Large Language Model which is based on the original BERT architecture. However, RoBERTa has the following modifications in the training procedure which make it more robust than the standard BERT models:

- The model is trained longer with larger batch sizes over more data.
- It is trained on longer sequences of data.
- BERT uses the task of Next Sentence Prediction (NSP) during pretraining. This task is not used while optimizing RoBERTa model.
- Dynamic masking techniques were applied that make the model more robust.
- BERT uses a Byte Pair Encoding vocabulary of $30k$ subword units which are learned after preprocessing the inputs. RoBERTa uses a larger Byte Pair Encoding Vocabulary of $50k$ subword units which are not preprocessed.

The variant of RoBERTa model that we use for fine-tuning on our task has also been pretrained on the Natural Language Inference task using the Multi-NLI [20] and Stanford-NLI [1] datasets. Natural

[1] `https://huggingface.co/`

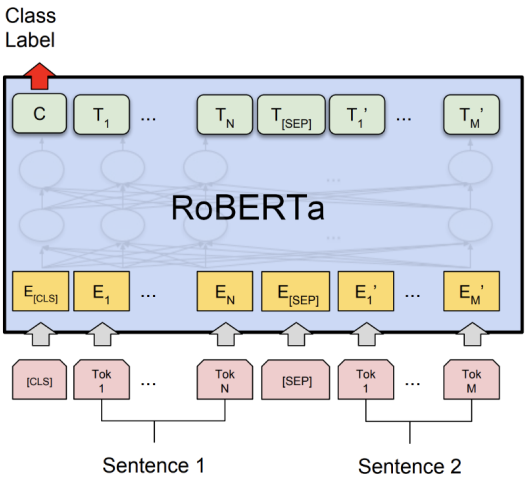

Figure 7: General Architecture of RoBERTa model used during pretraining

Language Inference task is the task of determining whether a given hypothesis is true (entailed in the given context), false or undetermined. These tasks have been shown to improve the performance of large language models on downstream Question Answering based tasks [2]. We have two types of accuracies that can be reported due to the architecture here:

- Instance Accuracy: This calculates whether the selected choice out of all the given choices was the correct one or not.
- Classification Accuracy: This calculates whether the model was correctly able to classify the given pair of question and choice or not.

4.2.2 BERT Small Pretrained on SQuAD

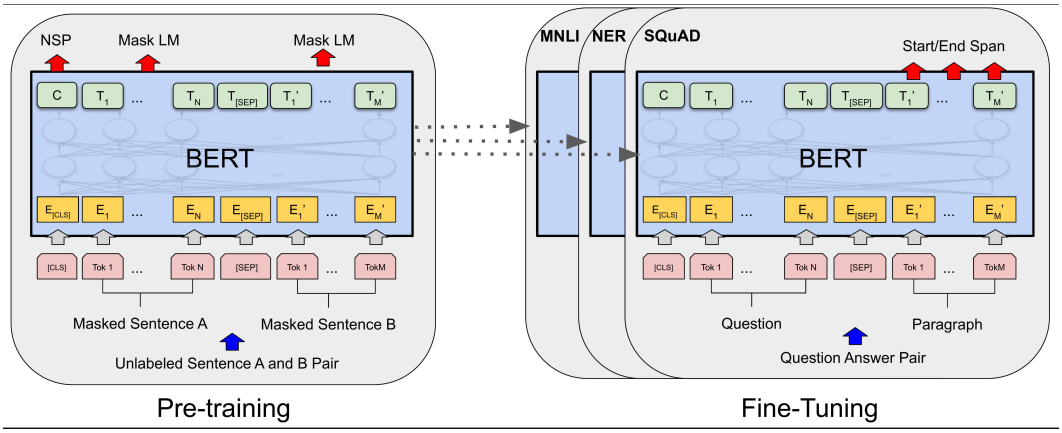

Figure 8: The original BERT Architecture

Bert small pretrained on SQuAD [15] uses a smaller version of BERT (Bidirectional Encoder Representations from Transformers)[5]. The architecture of BERT is shown here 8. Here is a breakdown of the key components of the BERT Architecture.

- Bidirectional: Unlike traditional language models that process text either from left-to-right or right-to-left, BERT processes the text in both directions simultaneously. This allows it to better understand the context of words in a sentence, as it takes into account both the preceding and the following words.

- Pre-training: BERT is first pre-trained on a large corpus of text using unsupervised learning. During this phase, BERT learns to predict missing words in a sentence (masked language modeling) and to determine whether two sentences follow each other (next sentence prediction). This pre-training enables BERT to learn general language understanding.
- Fine-tuning: After pre-training, BERT can be fine-tuned on a specific task, such as question-answering, sentiment analysis, or named entity recognition. In the case of the linked model, BERT has been fine-tuned on the SQUAD (Stanford Question Answering Dataset) dataset, which is designed for question-answering tasks. We used this model because SQUAD Dataset is similar to our science QA dataset [11].

Since it's a smaller version of BERT, it has fewer layers and parameters than the original BERT architecture. This makes the model faster and more efficient, although it may result in slightly lower performance compared to larger BERT models.

4.2.3 DistilRoBERTa with ResNet152: Incorporating Image context

Some questions also have an Image as part of the context. To incorporate the image into the model input, we have concatenated the image embedding with the text embedding of the question before passing it to the base model. The image encoder is a Convolutional Neural network. We have used the pre-trained ResNet-152 model in our encoder. The final layer of the ResNet-152 model is removed and replaced with a linear layer with 768 neurons. It is ensured that all the layers of the ResNet-152 model are frozen and their weights are not updated during training phase. The final replaced linear layer's weights will continue to be updated during training. The image embedding sizes is a tunable parameter and grid-search can be performed in order to find the optimal embedding size. For some questions, the hints provided were lengthy texts, but due to limited resources, we encountered memory errors. As a result, we had to trim the hints to reduce their length and continue training. The architecture for this model is shown in Figure 9

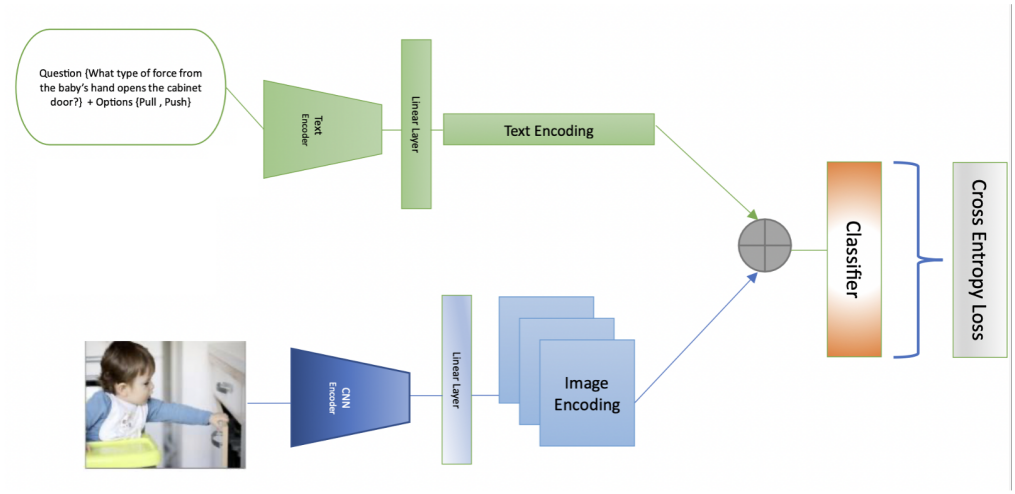

Figure 9: Architecture combining ResNet152 image embeddings with Distil Roberta text embeddings of the question

4.2.4 DistilRoBERTa with ResNet152: Incorporating Image and Hint context

Some questions have an Image along with hints (as text information) as part of the context. To incorporate the image and hint into the model input, we have concatenated the image embedding with the text embedding of the question and the text embedding of the hint before passing it to the base model. The image encoder is a Convolutional Neural network. We have used the pre-trained ResNet-152 model in our encoder. The final layer of the ResNet-152 model is removed and

replaced with a linear layer with 256 neurons. It is ensured that all the layers of the ResNet-152 model are frozen and their weights are not updated during training phase. The final replaced linear layer's weights will continue to be updated during training. The image embedding sizes is a tunable parameter and grid-search can be performed in order to find the optimal embedding size. The architecture for this model is shown in Figure 10

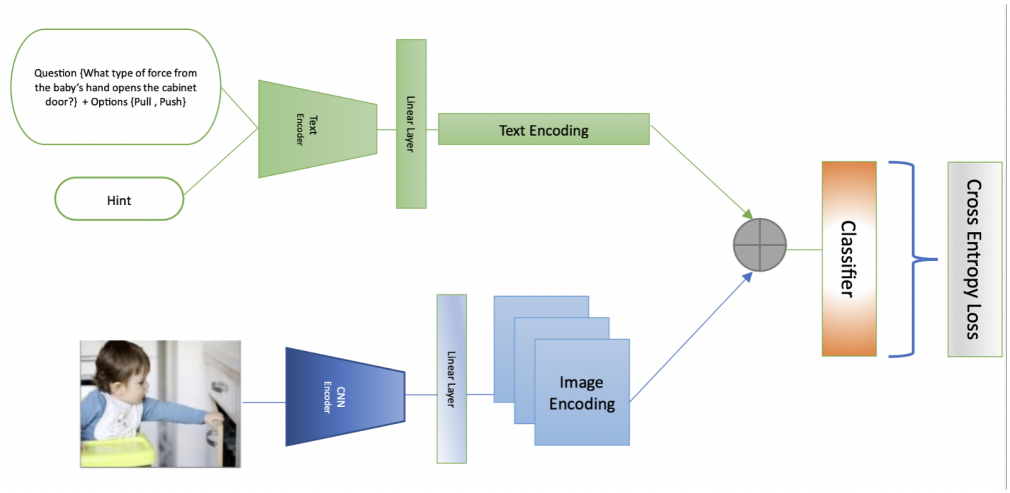

Figure 10: Architecture combining ResNet152 image embeddings with Distil Roberta question and hint text embeddings.

4.3 Siamese Transformer Model using Contrastive Loss

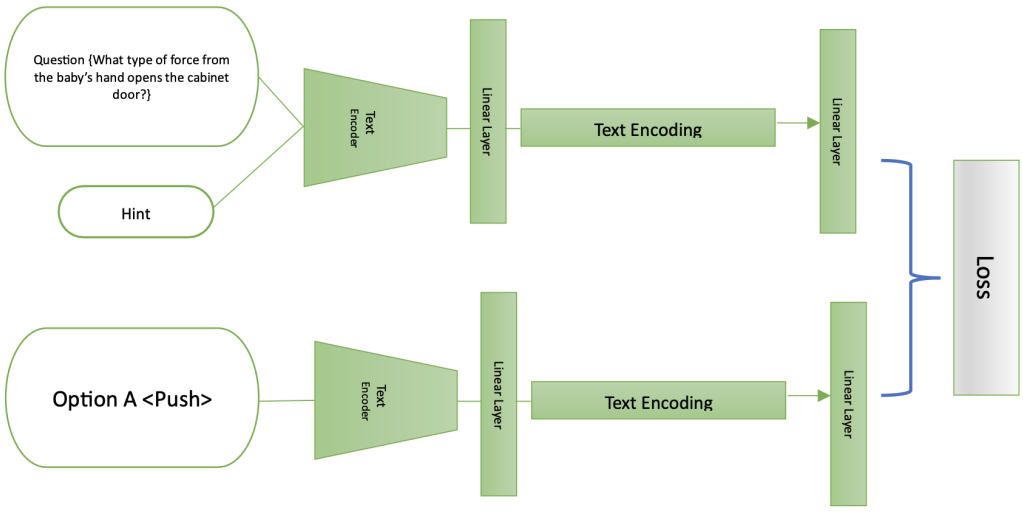

Figure 11: Architecture of our Siamese Transformer based model. These models use Constrastive or Margin Ranking Loss

The next architecture implemented consists of a siamese network with the towers made of pre-trained transformers using contrastive loss to answer multiple choice questions as shown in Figure 11 . In this architecture, we propose that intuitively the closest option to the question must be the

answer. We create a network using transformers to generate a fixed size embedding for the questions and a corresponding embedding for each of the choices. We then compute cosine similarity between the embedding of each choice and the question embedding. The choice which is closest to the question based on cosine similarity will be chosen as the answer. The model is experimented with a variety of pre-trained LLMs which can be used to generate embeddings. We first pass in the input tokens after padding to a fixed size into the LLM. We then generate embeddings for each of the input tokens and combine then using a mean/average operation to form a single embedding representing the question. We perform a similar operation to obtain a embedding for the individual choices. All the choice and question embeddings are of the same size. We then train this model using a contrastive loss. We define the contrastive loss to minimise the eucledian distance between the correct choice and the question while maximising the eucledian distance between the incorrect choice and question. Unlike the previous approach of classification, this approach takes the interaction between choices into consideration and there is no information loss from the choices. The approach helps the model generate richer embeddings where the question and the choice embeddings are similar for the correct choice and dissimilar for the incorrect choice.

$$L_c = \sum_{i=1}^{n} [y_i d_i^2 + (1 - y_i) \max(0, m - d_i)^2] \tag{2}$$

where n is the batch size, y_i is the binary label indicating whether the i-th example is a positive (1) or negative (0) example, d_i is the Euclidean distance between the question embedding and the i-th answer embedding (either correct or incorrect), m is the margin (i.e., the minimum distance that should separate positive and negative examples), and the loss is summed over all examples in the batch.

4.4 Siamese Transformer Model using Margin Ranking Loss (MR Loss)

In this subsection, we explore the previous architecture by modifying the loss function. We modify the contrastive loss and now use Margin Ranking Loss(Hinge Loss). Hinge loss is a loss function used in machine learning for training classifiers, particularly for binary classification problems (binary comparison) . The hinge loss measures the error between the similarity of question embedding with correct answer embedding (Sim_{QC}) and question embedding with incorrect answer embedding(Sim_{QI}) of our model. In the case of binary comparison, the hinge loss is defined as the maximum of 0 and the difference between the Sim_{QC} and Sim_{QI} separated by a margin. The margin is a hyperparameter that determines the minimum distance between the decision boundary and the data points.The motivation behind hinge loss is to penalize incorrect predictions that are not confidently predicted by the model. This is because the hinge loss function only applies a penalty to the model's output when the predicted score is less than the margin, encouraging the model to output scores that are far from the decision boundary.

In our usecase, we compare Sim_{QC} and Sim_{QI} using hinge loss. The case where $Sim_{QC} - Sim_{QI} >= m$, the model will incur no penalty. However, the case where $Sim_{QC} - Sim_{QI} < m$, the model incurs a penalty. The penalty forces the models to modify weights such that the question embedding and the correct answer are brought closer and the question embedding and the incorrect embedding will be further apart. This slowly tweaks the model to make better predictions. The equations 3, 4, 5 refer to the equations of the marginal ranking loss.

The figure 11 shows the model architecture. We see that the question embedding and the choice embedding are attained by passing the text through a transformer model. In our case we have used distilroberta-base. The embeddings then go through a linear layer to reduce it to an output dimension size of 100. The text encodings are then compared using cosine similarity. This results in similarity scores between each choice and the question. These scores are normalized through a softmax layer before they are passed into the Margin Ranking Loss function. Finally the margin ranking loss helps converge and the models predictive capabilities improve. During inference, we compare the similarity scores of all the chocies with the question embedding. The most similar choice (one with highest similarity score) is selected as the answer. We evaluate our models using accuracy as the prime evaluation metric.

$$Sim_{QC} = \frac{e^{CS_i/\tau}}{\sum_j e^{CS_j/\tau}} \tag{3}$$

$$Sim_{QI} = \frac{e^{IS_i/\tau}}{\sum_j e^{IS_j/\tau}} \tag{4}$$

$$L_c = max(0, -(Sim_{QC} - Sim_{QI}) + m) \tag{5}$$

5 Experimental Results

In this section, we discuss the experimental results of the models we trained. We run the experiments on various settings: with text, with question text and hint and finally question text , hint and image embeddings.

5.1 Experimental Settings

We consider the following the settings for our experiments which define what extra contextual features were used in addition to question and option pairs.

- **NO**: In this setting, no extra contextual features were given to the model.
- **TXT**: In this setting, text based hints were considered as the contextual features and passed to the model.
- **IMG**: In this setting, Image based features were also passed to the model via the pretrained ResNet encoder.

Due to resource constraints, we were only able to use the **IMG** experimental setting on select models.

5.2 Baseline Model : Siamese Network with LSTM and RNN units for Text Similarity

During the training phase, the training dataset is divided into batches of size 64 and Adam optimizer is being used with learning rate = 0.001. The model was trained for 25 epochs for both the LSTM and RNN Siamese models. The training and validation loss plots are shown in Figure **??** for the model with . The accuracy on test and validation dataset is reported in Table 3.

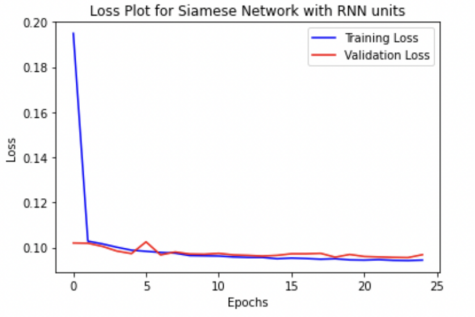

Figure 12: Loss Plot for Siamese Network with LSTM units

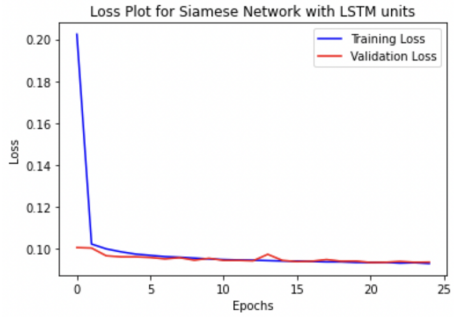

Figure 13: Loss Plot for Siamese Network with LSTM units

It can be seen from Figure 13, that the validation loss is decreasing and saturated at 0.093 loss value for the Siamese network with LSTM after 22 epochs. It can be seen that the accuracy of the model on the test and validation dataset is quite low for both the baseline models. The accuracy of the Siamese model with RNN units is much lower than the Siamese model with LSTM units. RNN models performs worse than LSTM in handling long-term dependencies because of the vanishing gradient problem, which causes the network to have difficulty retaining information from earlier time steps. LSTM, on the other hand, addresses this problem through the use of memory cells and gating

Dataset	Siamese LSTM Accuracy	Siamese RNN Accuracy
Validation	48.780%	34.659%
Test	52.028%	34.874%

Table 3: **Accuracy Report for Baseline Siamese Networks**

mechanisms, which allow it to selectively retain and discard information over time. Additionally, the baseline model has the lowest accuracy compared to other models. This is because the simple RNN/LSTM + fully connected layer architecture is not sufficient compared to the Transformer based architecture to establish similarity between question and answer text embeddings. Furthermore, incorporating hints and contextual information based on images can enhance the similarity between questions and answers, which is not included in our current model, and this may contribute to the low accuracy of the simple Siamese model.

5.3 Refactorization of Multi Choice Question Answering to Binary Question Answering

5.3.1 RoBERTa base NLI

The variant of RoBERTa model that we use for fine-tuning on our task has also been pretrained on the Natural Language Inference task using the Multi-NLI [20] and Stanford-NLI [1] datasets. Natural Language Inference task is the task of determining whether a given hypothesis is true (entailed in the given context), false or undetermined. These tasks have been shown to improve the performance of large language models on downstream Question Answering based tasks [2]. Figures 14 and 15 showcase the classification and instance accuracies for the RoBERTa-base-nli model. This model achieves an instance accuracy of 77.67%, classification accuracy of 83.93% on the test set. The final loss on the train set 0.2573 is while on the val set it is 0.3937 and on the test set it is 0.3934. This model was trained using a batch size of 8 and tuned using the ADAM optimizer.

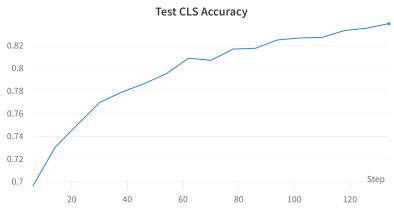

Figure 14: Classification Accuracy for Test dataset for RoBERTa-base-NLI

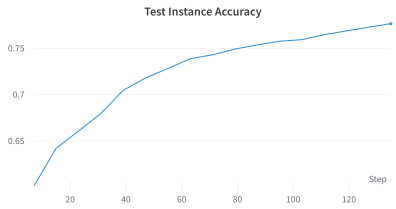

Figure 15: Instance Accuracy for Test dataset for RoBERTa-base-NLI

5.3.2 BERT Small Pretrained on SQuAD

This variant of BERT is a smaller version of BERT[5] which is pretrained on the SQuAD (Stanford Question Answering Dataset) [15]. We used this version of BERT as it is a lightweight model of BERT and SQuAD dataset is similar to the Science QA Dataset [11]. Figures 16 and 17 showcase the classification and instance accuracies for the Bert small pretrained on squad model. The red line represents the model which was trained without hints and the blue line represents the model which was trained with hints. This model achieves an instance accuracy of 72.01% when we use the hints and 70.9% when we don't use the hints. The classification accuracy of 80.008% on the test set is achieved when hints are used and 80.007% when hints are not used. We observed that test cls accuracy is almost the same for both the cases. This model was trained using a batch size of 8 and tuned using the ADAM optimizer.

5.3.3 DistilRoBERTa with ResNet152

To incorporate visual context into answer prediction, we utilized a pretrained ResNet152 network to obtain image embeddings. These image embeddings were concatenated with the text embeddings of

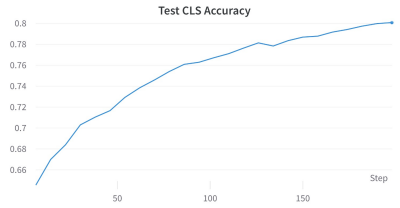

Figure 16: Classification Accuracy for Test dataset for Bert Small Pretrained on SQUAD

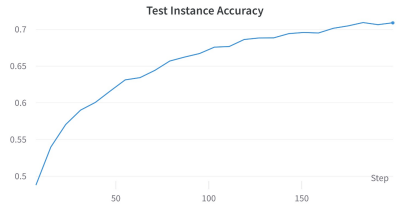

Figure 17: Instance Accuracy for Test dataset for BERT Small Pretrained on SQUAD

the questions from DistilRoBERTa model. The classification and instance accuracy is shown in Figures 18 and 19. Adam optimizer and a learning rate of $3e-6$ was used during the training phase. We incorporated binary cross entropy as the loss function for this model. This model achieved instance accuracy of 79.32% and classification accuracy of 84.54% after 100 epochs on the test dataset. The validation instance accuracy after 100 epochs was 79.39% and classification accuracy was 83.93%. The loss after 100 epochs on the test dataset was 0.6547 and 0.6278 on the validation dataset. However, due to RAM and GPU constraints during training, the ResNet152 layers were frozen, resulting in constant embeddings and no significant improvement in accuracy (as observed from Table 4). This can be attributed to the fact that the ResNet model was not pretrained to encode information specific to the downstream task of question answering. In the future, we plan to experiment with training the ResNet152 model to capture the necessary visual context for accurate answer prediction.

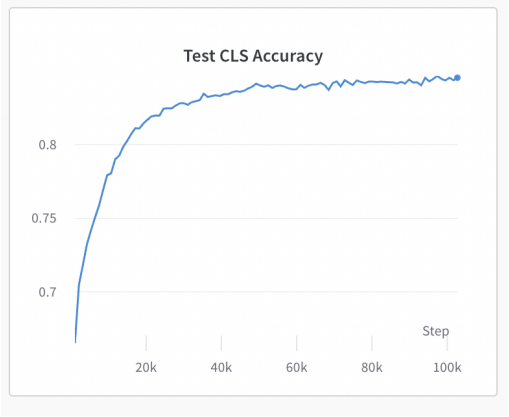

Figure 18: Classification Accuracy for Test dataset for ResNet152 with DistilRoBERTta

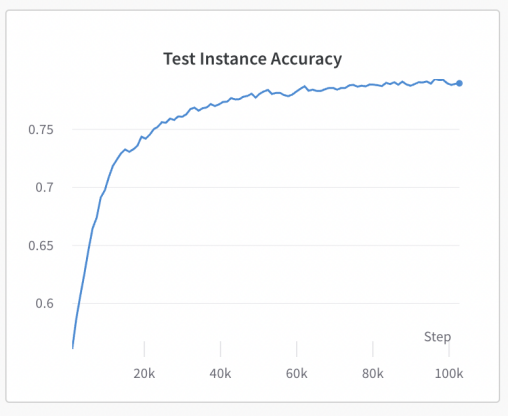

Figure 19: Instance Accuracy for Test dataset for ResNet152 with DistilRoBERTta

We also incorporated hints present for each question by concatenating the ResNet152 embeddings for the image along with the DistilRoBERTa embeddings for both the question and the hints. The classification and instance accuracy is shown in Figures 20 and 21. Adam optimizer and a learning rate of $3e-6$ was used during the training phase. We incorporated binary cross entropy as the loss function for this model. This model achieved instance accuracy of 78.02% and classification accuracy of 83.99% after 20 epochs on the test dataset. The validation instance accuracy after 100 epochs was 78.8% and classification accuracy was 83.65%. The loss after 20 epochs on the test dataset was 0.4584 and 0.4545 on the validation dataset. We noticed that some hints were excessively long, causing memory problems during training. To mitigate this issue, we truncated the hints to a maximum length of 100. Due to time and resource constraints we were unable to increase the number of training epochs for this model. We hypothesize that incorporating the complete hint context and increasing the number of epochs can enhance the test accuracy of the dataset, as the model would encompass all the essential context required for accurate answer prediction.

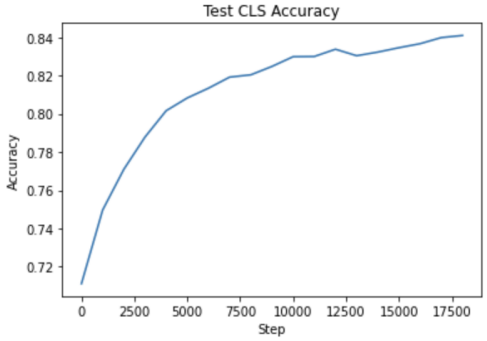

Figure 20: Classification Accuracy for Test dataset for ResNet152 with DistilRoBERTta (Question and Hint Context)

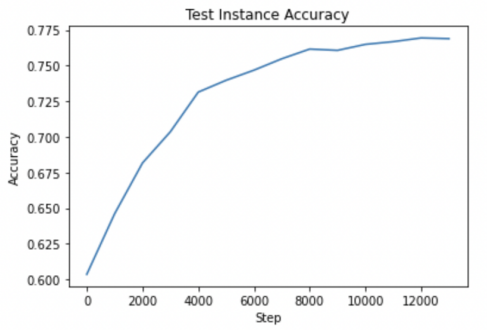

Figure 21: Instance Accuracy for Test dataset for ResNet152 with DistilRoBERTta (Question and Hint Context)

5.4 Contrastive Models

5.4.1 Siamese Transformer Model using Contrastive Loss

The Contrastive Loss model has shown a significant improvement in performance over other architectures. This could be attributed to the way the Contrastive Loss function measures the similarity between the embeddings of the question and the choices. It ensures that the embeddings of the correct choice are closer to the question embeddings as compared to the embeddings of the incorrect choices. This leads to better separation between the embeddings, making it easier for the model to classify the correct answer. The SiameseDistilRoberta with Contrastive Loss architecture also has the advantage of using pre-trained DistilRoberta embeddings which capture the semantic and contextual meaning of the text, making it easier for the model to understand the relationship between the question and the choices. However, the model still struggles to correctly classify some questions with complex language and requires further fine-tuning to improve performance. Overall, the SiameseDistilRoberta with Contrastive Loss architecture shows promising results and has the potential to be further improved upon. The results are presented in Table 4. The SiameseDistilRoberta model with Contrastive Loss achieves a training accuracy of 86.14%, validation accuracy of 79.2% and test accuracy of 78.7%. The learning rate used was $3e^{-6}$ using the ADAM optimizer and batch size was kept to 8. The test loss plot is shown in Figure 22 and the test instance accuracy plot is shown in Figure 23.

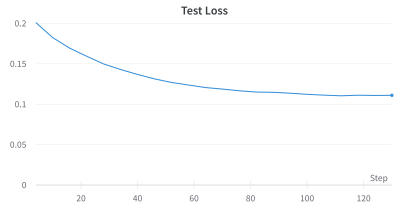

Figure 22: Contrastive Loss for Test dataset for SiameseDistilRoberta

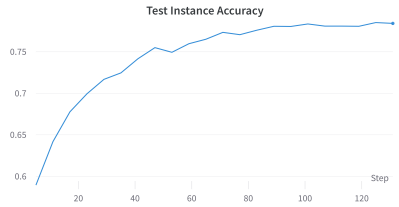

Figure 23: Instance Accuracy for Test dataset for SiameseDistilRoberta

5.4.2 Siamese Transformer Model using Margin Ranking Loss

In this architecture, we have experimented with the Margin Ranking Loss function which measures the similarity between the embeddings of the question and the correct choice and compares it with the similarity between the embeddings of the question and incorrect choice. This helps in incorporating the elimination mechanism which is often used by humans in Multiple Choice Question Answering tasks. This helps our models to learn better embeddings and perform better as compared to our other architectures. The results are presented in Table 4. The model with Margin Ranking Loss achieves training accuracy of 82.33%, validation accuracy of 81.97% and test accuracy of

Model	NO	TXT	IMG	Train Acc	Validation Acc	Test Acc
Siamese-RNN	✓				34.66%	34.87%
Siamese-LSTM	✓				48.78%	52.08%
BERT	✓			73.2%	71.2%	70.09%
BERT		✓		73.84%	72.88%	72.01%
DistilRoBERTa	✓			77.54%	77.01%	76.87%
DistilRoBERTa		✓		78.92%	78.83%	77.67%
DistilRoBERTa			✓	89.12%	79.39%	79.32%
DistilRoBERTa		✓	✓	85.82%	78.8%	78.02%
SiameseDistilRoberta - Contrastive Loss		✓		86.14%	79.2%	78.7%
SiameseDistilRoberta - MR Loss		✓		82.33%	81.97%	80.48%

Table 4: Instance Accuracy scores for QA models. **NO**=No-Context **TXT**=Text-Context and **IMG**=Image-Context

80.48%. The learning rate used was $3e^{-4}$ using the ADAM optimizer and batch size was kept to 8. The loss plots for the training, validation and test set are shown in the figure 24.

Figure 24: Training and Validation Loss Plots for Siamese Network with Margin Ranking Loss

6 Discussion

Based on the instance accuracy results obtained from training custom models on ScienceQA dataset, it can be observed that the Siamese models with pre-trained DistilRoberta architecture and fine-tuned with Contrastive and Margin Ranking Loss achieved the highest test accuracy of 78.7% and 80.48% respectively among models trained only on text context. One of main the reasons for the success of these contrastive models is due to the use of their respective loss functions, which are known to be powerful loss function for training siamese models. These loss functions help in learning better feature representations by maximizing the similarity between question and answer pairs that belong together while minimizing the similarity between pairs that do not belong together. Addition of images improved the test accuracy of the DistilRoberta model (Trained with Binary Cross Entropy loss) from 76.87% (with no context) to 79.32% with image context. Siamese LSTM achieved the second-highest validation accuracy of 48.78%, which is significantly lower than the best-performing model. This could be due to the fact that LSTM is not as effective as DistilRoberta in capturing complex linguistic structures and semantic relationships, which are crucial for answering questions. Similarly, Siamese RNN with the lowest accuracy of 34.66% might have suffered from the same limitations as LSTM. DistilRoberta for Binary Classification achieved reasonable accuracy of 77.01% on validation data and 76.87% on test data. However, this model does not take into account the fact that for a given question only one choice can be correct at a time. This could be a reason for the lower performance of this model compared to SiameseDistilRoberta models, which considers the embedding distances as well. Our best model performed better than the 2-shot GPT 3.5 model (Test accuracy 79.93%). This shows that simpler models with the right loss function can achieve equally good and sometimes better performance than few-shot heavily pre-trained models (Occam's Razor).

7 Conclusion

In conclusion, we have presented an extensive analysis of several deep learning models for Multiple Choice Question Answering on the ScienceQA dataset. Our analysis shows that SiameseDistilRoberta with Margin Ranking Loss is the most effective model for this task, achieving a test accuracy of 80.48%. We have also analyzed the impact of different model architectures and hyperparameters on the performance of these models. Our experiments reveal that pre-trained models, such as DistilRoberta, provide a strong baseline for this task.

However, we also observed that the performance of our models is limited by the complexity and the structure of the ScienceQA dataset. The dataset is highly heterogeneous in terms of the types of questions, and it includes a large number of questions that require external knowledge or domain-specific expertise to be answered correctly.

Overall, our findings suggest that Multiple Choice Question Answering remains a challenging task, and further research is needed to improve the performance of deep learning models on this task. We hope that our work will contribute to the development of more effective models and evaluation metrics for this task, and will encourage future research in this area.

8 Future Work

For future work, we can train the ResNet encoder as well to improve the image embedding specific to the question-answering context. We can also experiment with using Vision Transformers to encode the image context. We can also incorporate images in the SiameseDistilRoberta models to improve the performance of the model by providing more context of the question. Generating explanations along with the correct option improves explainability and interpretability of the model. This helps improves trust in a model so that it can be deployed in the real world. In the case of multiple choice questions, it can also help understand the chain-of-thought to arrive at the answer better. We can use auto-regressive models to generate explanations as well as part of future work.

References

[1] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015.

[2] Jifan Chen, Eunsol Choi, and Greg Durrett. Can nli models verify qa systems' predictions? *arXiv preprint arXiv:2104.08731*, 2021.

[3] Long Chen, Xin Yan, Jun Xiao, Hanwang Zhang, Shiliang Pu, and Yueting Zhuang. Counterfactual samples synthesizing for robust visual question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10800–10809, 2020.

[4] Garrison Cottrel. Knowledge base graph embedding module design for visual question answering model. *Slides on SimCLR*, page 108153, Winter 2023.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186, 2018.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[7] David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A Kalyanpur, Adam Lally, J William Murdock, Eric Nyberg, John Prager, Nico Schlaefer, and Chris Welty. Building watson: An overview of the deepqa project. *AI Magazine*, 31(3):59–79, 2010.

[8] Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *ACL*, pages 1601–1611, 2017.

[9] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16*, pages 121–137. Springer, 2020.

[10] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[11] Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[12] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. Ms marco: A human-generated machine reading comprehension dataset. In *NIPS*, 2016.

[13] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[14] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *EMNLP*, pages 2383–2392, 2016.

[15] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, 2016.

[16] Siva Reddy, Danqi Chen, and Christopher D. Manning. Coqa: A conversational question answering challenge. In *ACL*, pages 3548–3559, 2019.

[17] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. In *EMNLP*, 2017.

[18] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019.

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[20] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics, 2018.

[21] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6281–6290, 2019.

[22] Yinhan Zhang, Yi Tay, Dara Bahri, Donald Metzler, Da-Cheng Juan, Zhe Zhao, and Che Zheng. Reformulating task-transfer as low-resource meta-learning. In *ICLR*, 2020.

[23] Wenfeng Zheng, Lirong Yin, Xiaobing Chen, Zhiyang Ma, Shan Liu, and Bo Yang. Knowledge base graph embedding module design for visual question answering model. *Pattern recognition*, 120:108153, 2021.