AI-825 Mini Project Report

M Srinivasan
IMT2021058
IIIT-Bangalore

Siddharth Kothari
IMT2021019
IIIT-Bangalore

Sankalp Kothari
IMT2021028
IIIT-Bangalore

***Abstract*—This is the report for the Final Mini project for the course AI-825 (Visual Recognition). This project aims to create a model for Visual Question Answering (VQA). We have done two broad approaches for the same. Both approaches use ViT as the image feature extractor and BERT as the language feature extractor. One approach involves the use of a CoTRM layer to allow cross attention between the vit and bert embeddings, and then passes them through a feed forward neural network, while the other diretly passes the embeddings through the neural network.**

***Index Terms*—Multi-modal Transformers, Cross Attention, VQA.**

I. Introduction

The objective of this project is to create a model for visual question answering (VQA). VQA takes an image, and a question, and it is supposed to answer the question based on what is there in the image. The answer is to be determined from a predefined answer vocabulary. An example is provided in Figure 1.

Fig. 1. Question: Are there numbers on the clock face?

II. Dataset Acquistion

The VQA dataset contained a large set of images, questions and answers. Since it wasn't practical to work on such large data, we sampled 1/4th of the images from the dataset randomly and then picked out the corresponding questions and answers to the sampled images. We then created a tuple, that contains the image, question and corresponding answer to it. More than 89 % of the questions had single word answers to it, hence we curated the dataset in such a way that we will not have multiple word answers to a question and then the answer we choose for the question was based on the majority of answers given to that question.

III. Descriptions of the Models

A. The Vision and Language Transformer

In this section, we describe the architecture of our Visual Question Answering (VQA) system, which integrates a vision model (ViT) and a language model (BERT),

B. Language Model: BERT

BERT (Bidirectional Encoder Representations from Transformers) is a language representation model introduced by Devlin et al. [2] in 2018. BERT is pre-trained on a large corpus of text and is designed to understand the context of a word based on its surrounding words in a sentence. The model architecture consists of multiple transformer layers, where each layer applies self-attention to capture relationships between words.

In our VQA system, we use the `bert-base-uncased` model from the Hugging Face library. The BERT model takes a textual question as input and produces contextualized embeddings for each token in the question. The output from the BERT model is a fixed-size vector representing the overall context of the question.

C. Vision Model: ViT

Vision Transformer (ViT) is a model introduced by Dosovitskiy et al [1]. in 2020 that adapts the transformer architecture, originally designed for natural language processing, to image data. ViT splits an image into a sequence of patches, each treated as a token, and processes these tokens using transformer layers to capture spatial relationships.

In our VQA system, we use the `vit-base-patch16-224-in21k` model from the Hugging Face library. The ViT model takes an image as input, divides it into patches, and generates a sequence of

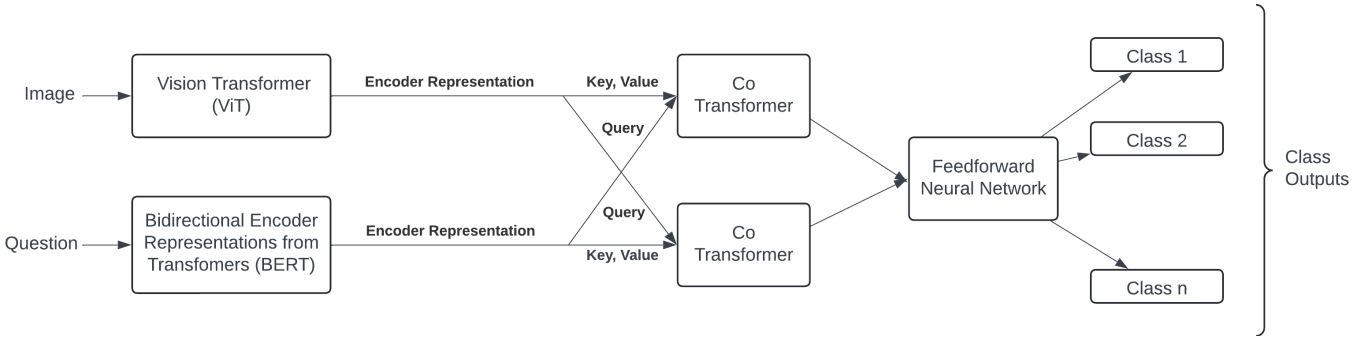

Fig. 2. Architecture for VQA with CoTRM

embeddings. The output from the ViT model is a fixed-size vector representing the visual information in the image.

D. With CoTRM

The CoTransformer (COTR) leverages attention mechanisms to fuse information from different modalities effectively.

The CoTransformer model is designed to capture the interactions between visual and textual inputs through multi-head attention mechanisms. It consists of multiple layers of CoTransformer blocks, each performing co-attention operations between the visual and textual embeddings. The key components of the COTR model include:

- **Multi-head Attention**: This component computes attention weights between the visual and textual embeddings, allowing the model to focus on relevant information from both modalities.
- **Linear Layers**: Linear transformation layers are used to project the input embeddings into a shared feature space, facilitating cross-modal interaction.
- **Layer Normalization**: To stabilize the training process and improve convergence, layer normalization is applied after each CoTransformer block.

The vectors from the two CoTransformers are concatenated to form a combined representation. The combined representation is passed through a fully connected layer with ReLU activation and dropout for regularization. The final output layer predicts the answer class.

The parameters are as follows - we get a 768 dimensional embedding as the output from the two CoTRM layers. We consider the CLS token's embeddings from both cases to get the output. These are passed through a neural network with hidden layer 1 having dimension as 512, and hidden layer 2 having dimension as 256.

The architecture is shown in Figure 2.

E. Without CoTRM

Our VQA model combines the outputs of the BERT and ViT models to answer questions based on visual input.

The model takes an image and a corresponding question as input. The ViT model extracts visual features from the image, resulting in a fixed-size vector. The BERT model extracts contextual embeddings from the question, resulting in a fixed-size vector.

The vectors from the ViT and BERT models are concatenated to form a combined representation. The combined representation is passed through two fully connected layers with ReLU activations and dropout for regularization. The final output layer predicts the answer class.

The parameters are as follows - we get a 768 dimensional embedding as the output from ViT and BERT for each token. We consider the CLS token's embeddings from both cases to get the output. These are passed through a neural network with hidden layer 1 having dimension as 512, and hidden layer 2 having dimension as 256.

The architecture is shown in Figure 3.

IV. Experiments

We have carried out the following experiments, whose results are noted in the next section.

1) We first carried out training of both the models (with and without CoTRM), while keeping the ViT and BERT models as constant (i.e. without fine-tuning those). The Batch size used was 32, while the number of batches considered in one epoch was 30. This was done to avoid the OOM, which occurs with larger batch sizes.
2) In the second experiment, we tried to check whether increasing the values of the number of training examples considered for a gradient propagation affects the accuracy or not. For this, we did the first experiment again, with an increase in the
3) In the third experiment, we then tried to train the ViT and BERT models as well, along with our own models, without using LoRA. We intend to capture the time taken to fine tune the models in this case.
4) For the fourth experiment, we tried to incorporate a learning rate scheduler into the model, but without using the LoRA fine-tuning methodology. It dynamically changes the learning rate as the epochs progress. It is a linear scheduler where the learning rate ratio decays linearly from 1 to 0.1 over the course of 10 epochs.
5) For the last experiment, we incorporated the LoRA fine-tuning in the model. We use LoRA to fine tune the linear layers in the ViT and BERT models, each with rank 10.

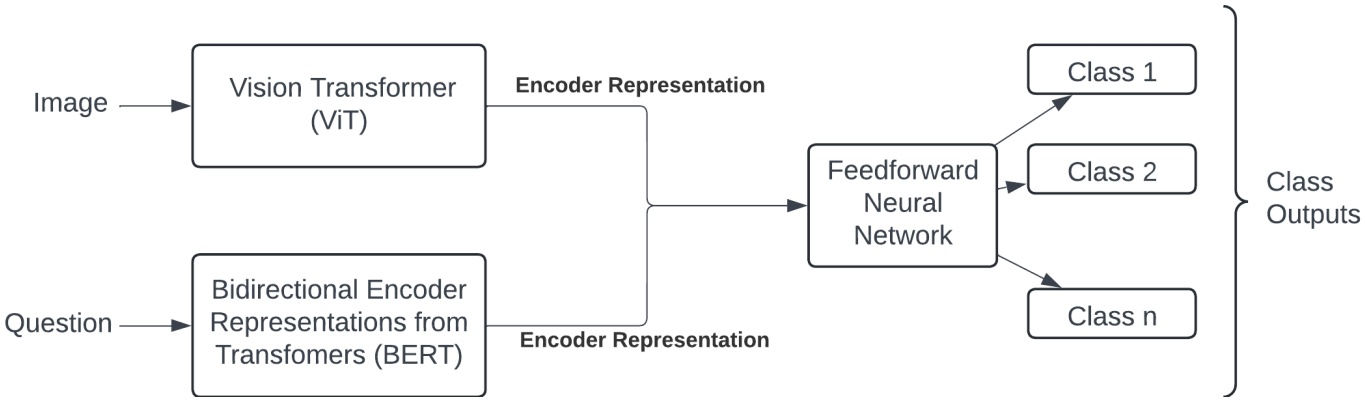

Fig. 3. Architecture for VQA without CoTRM

TABLE I
WITHOUT COTRM - NOT TRAINING VIT AND BERT

Epoch Number	Metrics							
	Train Accuracy	*Train Precision*	*Train Recall*	*Train F1 Score*	*Val Accuracy*	*Val Precision*	*Val Recall*	*Val F1 Score*
1	15.1042	0.0321	0.151	0.053	19.5833	0.0384	0.1958	0.0641
2	19.7917	0.0725	0.1979	0.0966	17.3958	0.0303	0.174	0.0516
3	18.6458	0.0677	0.1865	0.0979	22.0833	0.0488	0.2208	0.0799
4	19.5833	0.0891	0.1958	0.1051	18.75	0.0706	0.1875	0.0951
5	18.8542	0.07	0.1885	0.102	17.9167	0.0321	0.1792	0.0544
6	19.6875	0.0727	0.1969	0.1057	18.0208	0.0325	0.1802	0.055
7	19.7917	0.0738	0.1979	0.1075	19.5833	0.0748	0.1958	0.1037
8	18.6458	0.0709	0.1865	0.1023	18.9583	0.0359	0.1896	0.0604
9	16.5625	0.0604	0.1656	0.0885	21.875	0.0843	0.2188	0.1217
10	18.3333	0.0688	0.1833	0.0996	19.6875	0.0833	0.1969	0.1106

TABLE II
WITH COTRM - NOT TRAINING VIT AND BERT

Epoch Number	Metrics							
	Train Accuracy	*Train Precision*	*Train Recall*	*Train F1 Score*	*Val Accuracy*	*Val Precision*	*Val Recall*	*Val F1 Score*
1	18.65	0.0652	0.1865	0.0956	19.38	0.0378	0.1938	0.0633
2	17.40	0.0680	0.1740	0.0946	17.29	0.0299	0.1729	0.0510
3	19.17	0.0690	0.1917	0.1010	18.02	0.0678	0.1802	0.0667
4	18.13	0.0690	0.1813	0.0999	17.19	0.0295	0.1719	0.0504
5	20.21	0.0784	0.2021	0.1128	18.75	0.0352	0.1875	0.0592
6	22.08	0.0888	0.2208	0.1259	20.21	0.0408	0.2021	0.0679
7	20.31	0.0751	0.2031	0.1093	17.81	0.0317	0.1781	0.0539
8	19.38	0.0747	0.1938	0.1078	17.92	0.0321	0.1792	0.0544
9	19.27	0.0737	0.1927	0.1066	20.21	0.0408	0.2021	0.0679
10	18.75	0.0720	0.1875	0.1041	19.90	0.0396	0.1990	0.0660

TABLE III
WITHOUT COTRM - TRAINING VIT AND BERT WITHOUT LORA

Epoch Number	Metrics							
	Train Accuracy	*Train Precision*	*Train Recall*	*Train F1 Score*	*Val Accuracy*	*Val Precision*	*Val Recall*	*Val F1 Score*
1	14.1667	0.072	0.1417	0.0818	18.0208	0.0325	0.1802	0.055
2	19.5833	0.0723	0.1958	0.101	20.4167	0.0417	0.2042	0.0692
3	19.2708	0.0831	0.1927	0.1161	17.6042	0.0528	0.176	0.0812
4	18.9583	0.149	0.1896	0.1643	20.3125	0.0874	0.2031	0.1183
5	21.875	0.192	0.2188	0.1977	21.4583	0.0916	0.2146	0.1235
6	23.75	0.2157	0.2375	0.2196	23.125	0.1079	0.2313	0.1462
7	21.875	0.2012	0.2188	0.204	20.7292	0.085	0.2073	0.1185
8	20.1042	0.177	0.201	0.1796	21.3542	0.0954	0.2135	0.1291
9	21.25	0.1934	0.2125	0.1979	22.0833	0.1019	0.2208	0.1387
10	21.0417	0.1883	0.2104	0.1909	23.5417	0.1077	0.2354	0.1454

TABLE IV
WITH COTRM- TRAINING VIT AND BERT WITHOUT LORA

Epoch Number	Metrics							
	Train Accuracy	*Train Precision*	*Train Recall*	*Train F1 Score*	*Val Accuracy*	*Val Precision*	*Val Recall*	*Val F1 Score*
1	18.020833	0.077603	0.180208	0.107664	21.25	0.092962	0.2125	0.127713
2	20.104167	0.159917	0.201042	0.176013	20.3125	0.186664	0.203125	0.130954
3	20.729167	0.181385	0.207292	0.188151	21.875	0.105829	0.21875	0.140157
4	23.229167	0.214222	0.232292	0.216501	21.770833	0.089566	0.217708	0.12235
5	23.333333	0.214223	0.233333	0.217413	21.979167	0.098065	0.219792	0.132185
6	23.541667	0.201018	0.235417	0.209058	24.479167	0.10608	0.244792	0.145348
7	22.395833	0.198244	0.223958	0.200775	21.5625	0.184398	0.215625	0.133054
8	24.479167	0.210791	0.244792	0.218261	23.645833	0.110136	0.236458	0.147805
9	25.0	0.221004	0.25	0.225418	24.375	0.109099	0.24375	0.148666
10	22.291667	0.206925	0.222917	0.203265	26.5625	0.131002	0.265625	0.171796

TABLE V
WITHOUT COTRM - TRAINING VIT AND BERT WITH SCHEDULER

Epoch Number	Metrics							
	Train Accuracy	*Train Precision*	*Train Recall*	*Train F1 Score*	*Val Accuracy*	*Val Precision*	*Val Recall*	*Val F1 Score*
1	18.020833	0.077603	0.180208	0.107664	21.25	0.092962	0.2125	0.127713
2	20.104167	0.159917	0.201042	0.176013	20.3125	0.186664	0.203125	0.130954
3	20.729167	0.181385	0.207292	0.188151	21.875	0.105829	0.21875	0.140157
4	23.229167	0.214222	0.232292	0.216501	21.770833	0.089566	0.217708	0.12235
5	23.333333	0.214223	0.233333	0.217413	21.979167	0.098065	0.219792	0.132185
6	23.541667	0.201018	0.235417	0.209058	24.479167	0.10608	0.244792	0.145348
7	22.395833	0.198244	0.223958	0.200775	21.5625	0.184398	0.215625	0.133054
8	24.479167	0.210791	0.244792	0.218261	23.645833	0.110136	0.236458	0.147805
9	25.0	0.221004	0.25	0.225418	24.375	0.109099	0.24375	0.148666
10	22.291667	0.206925	0.222917	0.203265	26.5625	0.131002	0.265625	0.171796

TABLE VI
WITH COTRM - TRAINING VIT AND BERT WITH SCHEDULER

Epoch Number	Metrics							
	Train Accuracy	*Train Precision*	*Train Recall*	*Train F1 Score*	*Val Accuracy*	*Val Precision*	*Val Recall*	*Val F1 Score*
1	18.4375	0.0769	0.1844	0.1083	21.3542	0.0456	0.2135	0.0752
2	20.1042	0.0791	0.201	0.1129	18.8542	0.0355	0.1885	0.0598
3	19.0625	0.0706	0.1906	0.1012	19.5833	0.0384	0.1958	0.0641
4	18.4375	0.0691	0.1844	0.1006	20.2083	0.0408	0.2021	0.0679
5	20.5208	0.0743	0.2052	0.108	18.5417	0.0344	0.1854	0.058
6	20.3125	0.0764	0.2031	0.111	16.9792	0.0288	0.1698	0.0493
7	20.7292	0.0714	0.2073	0.0933	17.5	0.0306	0.175	0.0521
8	18.6458	0.0684	0.1865	0.0988	19.4792	0.0379	0.1948	0.0635
9	20.4167	0.0818	0.2042	0.1167	17.3958	0.0303	0.174	0.0516
10	18.8542	0.0746	0.1885	0.1068	17.0833	0.0292	0.1708	0.0499

TABLE VII
WITH COTRM - TRAINING VIT AND BERT WITH LORA

Epoch Number	Metrics							
	Train Accuracy	*Train Precision*	*Train Recall*	*Train F1 Score*	*Val Accuracy*	*Val Precision*	*Val Recall*	*Val F1 Score*
1	19.4792	0.0891	0.1948	0.1176	19.1667	0.0367	0.1917	0.0617
2	18.5417	0.0663	0.1854	0.0974	18.4375	0.034	0.1844	0.0574
3	18.5417	0.0687	0.1854	0.0992	18.8542	0.0355	0.1885	0.0598
4	18.75	0.0704	0.1875	0.1024	19.2708	0.0371	0.1927	0.0623
5	19.0625	0.0741	0.1906	0.1037	18.6458	0.0348	0.1865	0.0586
6	19.6875	0.0765	0.1969	0.1095	20.625	0.0425	0.2062	0.0705
7	19.375	0.0743	0.1938	0.1074	18.5417	0.0344	0.1854	0.058
8	19.7917	0.0729	0.1979	0.1042	19.7917	0.0392	0.1979	0.0654
9	18.6458	0.0725	0.1865	0.1043	19.4792	0.0379	0.1948	0.0635
10	16.1458	0.0571	0.1615	0.0835	18.8542	0.0355	0.1885	0.0598

TABLE VIII
WITHOUT COTRM - TRAINING VIT AND BERT WITH LORA

Epoch Number	Metrics							
	Train Accuracy	*Train Precision*	*Train Recall*	*Train F1 Score*	*Val Accuracy*	*Val Precision*	*Val Recall*	*Val F1 Score*
1	16.56%	0.0651	0.1656	0.0935	19.69%	0.0388	0.1969	0.0648
2	18.02%	0.0675	0.1802	0.0982	19.90%	0.0396	0.1990	0.0660
3	18.85%	0.0702	0.1885	0.1021	21.25%	0.0452	0.2125	0.0745
4	18.02%	0.0673	0.1802	0.0973	20.00%	0.0400	0.2000	0.0667
5	18.23%	0.0656	0.1823	0.0958	18.75%	0.0352	0.1875	0.0592
6	18.13%	0.0684	0.1813	0.0984	19.38%	0.0375	0.1938	0.0629
7	17.71%	0.0624	0.1771	0.0895	20.94%	0.0438	0.2094	0.0725
8	19.38%	0.0695	0.1938	0.0962	18.02%	0.0325	0.1802	0.0550
9	17.81%	0.0636	0.1781	0.0928	15.21%	0.0231	0.1521	0.0402
10	20.52%	0.0829	0.2052	0.1169	17.40%	0.0303	0.1740	0.0516

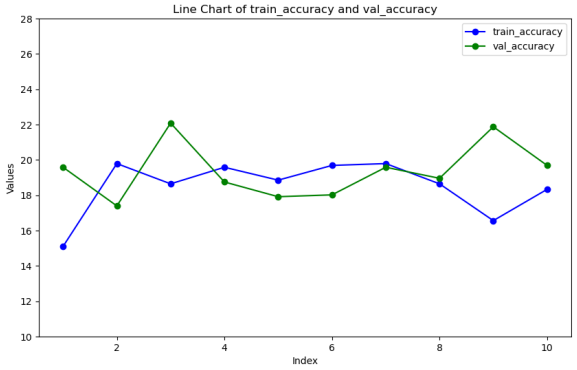

Fig. 4. Accuracy vs epochs for the without CoTRM model using pretrained ViT and BERT models

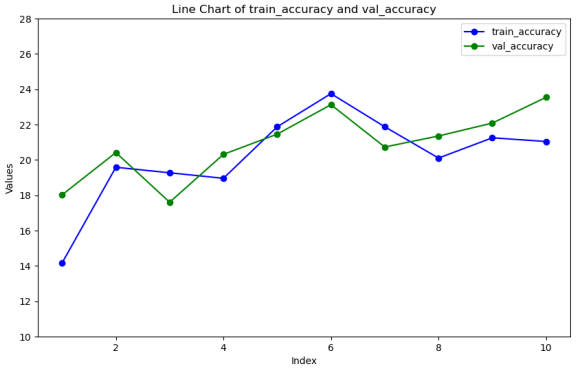

Fig. 6. Accuracy vs epochs for the without CoTRM model using finetuned ViT and BERT models

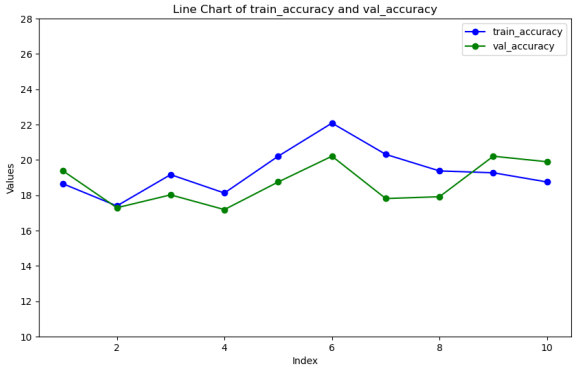

Fig. 5. Accuracy vs epochs for the CoTRM model using pretrained ViT and BERT models

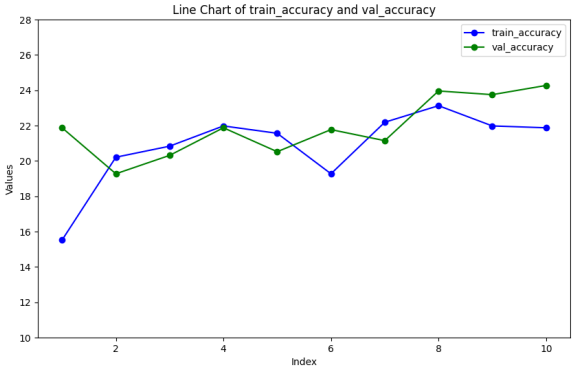

Fig. 7. Accuracy vs epochs for the CoTRM model using finetuned ViT and BERT models

V. CHALLENGES FACED

Since we had very less time for fine tuning and testing our model, the accuracy reported by our model is lower. We feel given more time and more GPU time to train models, we would be able to improve on the accuracy further.

Working with the VQA dataset wasn't very trivial. The dataset size was approx 20Gb and downloading that directly wasn't possible. We had to find other alternatives to download the dataset and then do the sampling of the images. Even after doing the sampling, the size of our dataset was approx 3-4GB.

VI. RESULTS

We have captured the following metrics for all the four types of experiments - accuracy, across all the epochs, along with precision, recall values and f1 scores, for all experiments except the second type of experiments. For experiment 2,

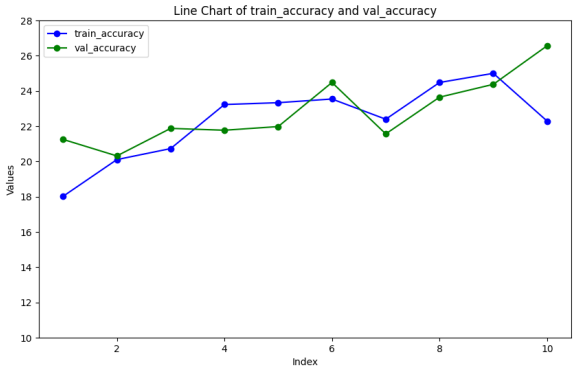

Fig. 8. Accuracy vs epochs for the CoTRM model using finetuned ViT and BERT models with LORA scheduler

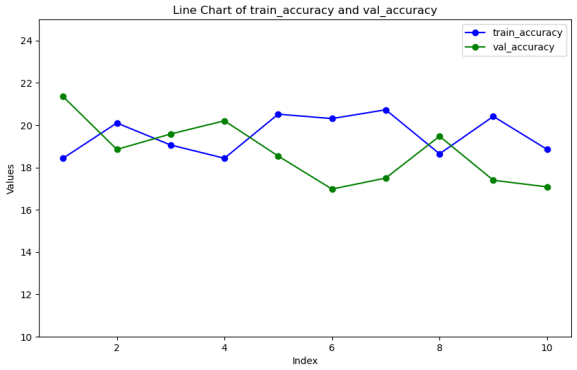

Fig. 9. Accuracy vs epochs for the CoTRM model with LoRA scheduler

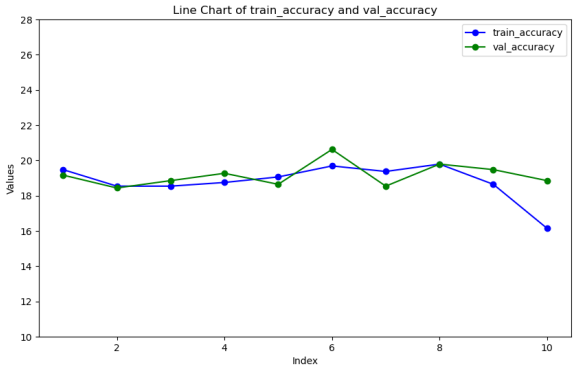

Fig. 10. Accuracy vs epochs for the CoTRM model with LoRA finetuning

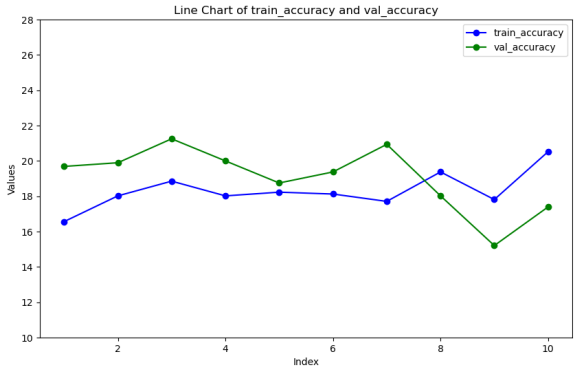

Fig. 11. Accuracy vs epochs for the without CoTRM model with LoRA finetuning

we've included the accuracies across the epochs. We have also included the accuracy vs epoch plots for experiments 1,3,4,5.

We use the following abbreviations to denote models in the table for training times -

- W/o - without
- CoTRM - Co Transformer
- The numbers indicate the number of batches per epoch, and the batch size
- No - The models where ViT and BERT were not fine-tuned.
- NoLoRA - The models where ViT and BERT were trained without using LoRA.
- Sch - The models where ViT and BERT were trained without using LoRA, but with the learning rate scheduler.
- LoRA - The models where ViT and BERT were trained using LoRA.

TABLE IX
WITHOUT COTRM - NOT TRAINING VIT AND BERT, INCREASED BATCH SIZE

Epoch Number	**Metrics**	
	Train Accuracy	***Val Accuracy***
1	17.239583333333332	19.427083333333332
2	18.229166666666664	19.140625
3	19.322916666666668	19.088541666666668
4	18.880208333333336	18.541666666666668
5	18.932291666666668	19.6875
6	18.776041666666668	18.307291666666668
7	18.359375	18.932291666666668
8	18.880208333333336	19.635416666666668

VII. CONCLUSIONS

1) There could be a better way of preprocessing the data. There is a large variety of questions, ranging from Qualitative questions such as - "What color is the umbrella", to more complicated numerical questions such as "How many zebras are there in the image"; from simple yes/no based questions, to almost impossible questions such as

TABLE X
WITH COTRM - NOT TRAINING VIT AND BERT, INCREASED BATCH SIZE

Epoch Number	Metrics	
	Train Accuracy	*Val Accuracy*
1	17.552083333333332	18.333333333333332
2	19.244791666666668	18.776041666666668
3	19.53125	19.21875
4	19.713541666666668	19.375
5	19.192708333333332	18.203125
6	18.932291666666668	19.192708333333332
7	19.166666666666668	19.296875
8	19.557291666666668	18.203125
9	18.697916666666668	18.828125
10	18.932291666666668	18.984375

TABLE XI
TRAINING TIMES VS EPOCHS FOR EACH MODEL

Model Name	Number of Epochs	Training Time
W/o CoTRM 32,30 No	10	1 hours 50 minutes
CoTRM 32,30 No	10	2 hours
W/o CoTRM 64,60 No	8	7 hours 30 minutes
CoTRM 64,60 No	10	9 hours 10 minutes
W/o CoTRM 32,30 NoLoRA	10	2 hours 45 minutes
CoTRM 32,30 NoLoRA	10	2 hours 55
W/o CoTRM 32,30 Sch	10	2 hours 35 minutes
CoTRM 32,30 Sch	10	2 hours 40 minutes
W/o CoTRM 32,30 LoRA	10	2 hours 20 minutes
CoTRM 32,30 LoRA	10	2 hours 25 minutes

- "What time is it on the watch", or "Who made the clock".

2) This variety of questions may indicate the need for a better preprocessing than what we have done. Ours is very simplistic - make a triple of image, question and answer pairs, and run the model to predict the answer. There may be better ways of handling the types of questions.
3) We have taken the answer to be the majority of the answers provided by the 10 attendants. However, it may be fruitful in certain cases to consider the variation in the answers as separate data points. We felt it may heavily bias the model to give higher accuracies if the common consensus for that question is the same. Hence we decided against it.
4) The key may also be in the CoTRM layers being more dense, i.e. instead of their being only one self attention layer, there may be multiple encoders stacked on top of one another. Training this may be difficult but it may yield better accuracy.
5) The accuracy over all the models and experiments remained roughly the same, ranging from 19% to 23%. This may be in part due to our preprocessing and due to the limited number of examples we actually ended up using. The V1A dataset is very vast, and we used only 20000 images, out of which in an epoch we only considered roughly 1000 images. This may result in many images not even being considered during the training process, on which the model is amost surely going to perform poorly.
6) We do observe, as expected, an increase in the accuracy when we allow the fine tuning of both the ViT and BERT models, as compared to when we allowed it to be constant. The magnitude varies. The training time, again as expected, is the least for when the ViT and BERT are not fine tuned, then for when they are fine tuned using LoRA, followed by when they are done using the scheduler, or without the scheduler.

RELEVANT LINKS

1) The link to the model parameters and training and validation results can be found here - Model Weights
2) The Github Link to the project can be found here - Github
3) The Link to download the datasets can be found here - Datasets

REFERENCES

[1] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S. and Uszkoreit, J., 2020. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.
[2] Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
[3] Lu, Jiasen, Dhruv Batra, Devi Parikh, and Stefan Lee. "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks." Advances in neural information processing systems 32 (2019).
[4] Pytorch documentation - link
[5] Numpy documentation - link
[6] Pandas documentation - link
[7] Matplotlib documentation - link
[8] Scikit Learn documentation - link
[9] HuggingFace ViT model - link
[10] HuggingFace BERT - link