Visual Question Answering on Real World Images

Wahaj Ahmed Butt (372890) - Model Training and Experimentation
Muhammad Suhaib Aslam (378332) - Data Preprocessing and Model Creation
Muhammad Bashar Siddiqui (366006) - Testing and Inference
SEECS, NUST
Islamabad, Pakistan

***Abstract*—This report presents a model for Visual Question Answering (VQA) related to Real World Images by combining convolutional neural networks (CNNs) and transformer-based language models. Our architecture integrates ResNet50 for image feature extraction and BERT for question encoding. We evaluate our model on the processed DAQUAR and analyze its performance using both accuracy and semantic similarity metrics.**

***Index Terms*—Visual Question Answering, BERT, ResNet50, Deep Learning, Transformers, Semantic Similarity**

I. Introduction

Visual Question Answering (VQA) is a complex task that involves understanding and reasoning about visual content in conjunction with natural language questions. The task demands the integration of vision and language processing to produce accurate answers. This work aims to leverage the capabilities of BERT for language understanding and ResNet50 for image feature extraction to develop an effective VQA model that can be used for Real World Images - particularly Indoor ones.

II. Related Work

Previous approaches to VQA have explored various neural architectures, including joint embeddings of image and text features, attention mechanisms, and transformer models. The VGG19, InceptionV3, and EfficientNetB2 architectures have been utilized for their strong performance in image feature extraction. Additionally, recurrent neural networks like LSTM and GRU have been integrated to handle the sequential nature of textual data. Vision-Language Transformers (ViLT) represent a more recent advancement, combining vision and language processing within a unified transformer framework, demonstrating significant improvements in VQA tasks.

III. Dataset Chosen

We utilize the Processed DAQUAR dataset for our project, which contains a diverse set of images paired with questions and corresponding answers - particularly 1,500 images and 25,000 total question-answer pairs. Breaking down into 6794 training and 5674 test question-answer pairs, this dataset draws from images sourced from the NYU-Depth V2 Dataset and averages about 9 Q&A pairs per image. It is basically a refined rendition of the Full DAQUAR Dataset, with questions standardized for enhanced compatibility with tokenizers. The image IDs, questions, and answers are organized in a tabular (CSV) format, facilitating straightforward loading and utilization for VQA model training.

IV. Model Architecture

Our proposed Visual Question Answering (VQA) model, as seen in 1, comprises the ResNet50 and BERT models.

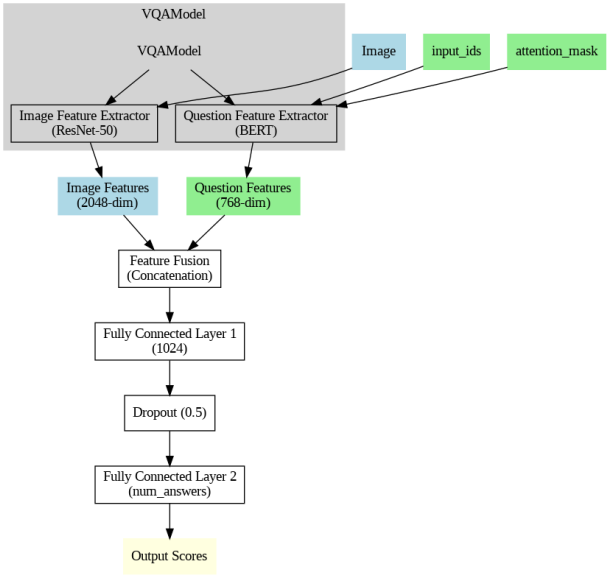

Fig. 1. Model Architecture

To that end, it consists of three primary components: image feature extraction, question encoding, and feature fusion and classification.

A. *Image Feature Extraction*

The image features are extracted using a pretrained ResNet50 model. Specifically, we replace the final classification layer of ResNet50 with an identity layer, which results in a 2048-dimensional feature vector for each image. This modification allows the model to leverage the pretrained convolutional layers of ResNet50 for robust feature extraction while omitting the task-specific classification layer.

B. *Question Encoding*

For encoding the textual questions, we employ the BERT model (Bidirectional Encoder Representations from Transformers). BERT is pretrained on a vast corpus of text, providing rich contextual embeddings. For our VQA task, we fine-tune BERT, using the output corresponding to the [CLS] token. This output is a 768-dimensional feature vector that captures the contextual representation of the input question.

C. Feature Fusion and Classification

To integrate the extracted image and question features, we concatenate the 2048-dimensional image feature vector from ResNet50 with the 768-dimensional question feature vector from BERT, resulting in a combined feature vector of size 2816. This combined vector is then processed through a fully connected layer (FC1) to reduce the dimensionality to 1024. To mitigate overfitting, a dropout layer is applied with a dropout rate of 0.5. Finally, the output is fed into another fully connected layer (FC2) to produce the answer logits, which correspond to the possible answers.

The overall architecture can be summarized as follows:

- Image Feature Extraction: ResNet50 (2048-dimensional features)
- Question Encoding: BERT (768-dimensional features)
- Feature Fusion and Classification: Concatenation, FC1 (1024-dimensional), Dropout, FC2 (answer logits)

V. Experimentation

To train our model effectively, we used a variety of hyperparameters and techniques, which are detailed below.

A. Hyperparameters

- Batch Size: We experimented with batch sizes of 16, 32, 64, and 128. The optimal performance was observed with a batch size of 64, balancing memory usage and training speed.
- Learning Rate: We used an initial learning rate of 1e-4 with a cosine annealing schedule to gradually decrease the learning rate, helping the model to converge smoothly.
- Optimizer: Adam optimizer was chosen for its effectiveness in handling sparse gradients on noisy problems.
- Epochs: The model was trained for 30 epochs, with early stopping based on validation loss to prevent overfitting.
- Dropout Rate: A dropout rate of 0.5 was applied to the fully connected layer to prevent overfitting.
- Weight Decay: A weight decay of 1e-5 was used to regularize the model and prevent overfitting.

B. Additional Tuning

- Mixed Precision Training: This technique was employed to reduce memory usage and speed up the training process by using half-precision floating-point numbers where possible.
- Gradient Clipping: Gradient clipping was applied with a maximum norm of 1.0 to prevent exploding gradients and ensure stable training.
- Num Workers: Data loading was parallelized using 4 workers to speed up the data preprocessing pipeline.
- Semantic Similarity: Using the ”paraphrase-MiniLM-L6-v2” sentence-transformer, we used the SBERT similarity metric to measure the relation and closeness between the predicted and actual answers.

C. Loss and Accuracy Metrics

We employed CrossEntropyLoss as our loss function, defined as:

$$\text{Loss} = -\sum_{i=1}^{N} y_i \log(\hat{y}_i) \tag{1}$$

where y_i represents the true label and $\hat{y}_i$ denotes the predicted probability for class i.

Accuracy was computed using the formula:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \tag{2}$$

These metrics provide a comprehensive evaluation of the model's performance, highlighting its ability to accurately classify and understand visual questions.

D. Testing Methadology

In order to test our trained model on real-world data, we devised a GUI using Gradio. By deploying it over Google Colab, we gained the capability to input an image of our own choice to the model and ask it questions question regarding said image - hence testing it on never-before-seen images.

VI. Results and Analysis

Our model's performance was rigorously evaluated across multiple metrics, with a primary focus on accuracy and semantic similarity. The results indicated a notable improvement in both training and validation metrics throughout the training process.

A. Training and Validation Performance

The initial training phase started with a training loss of 5.1493 and a training accuracy of 8.715%. Over the course of 30 epochs, the model exhibited significant improvement, with the training loss reducing to 3.3390 and the training accuracy rising to 34.54%. The validation accuracy, which began at 18.99%, ultimately reached 54.90% by the end of the training phase.

Fig. 2. Training and Validation Loss and Accuracy

Figure 2 illustrates the progression of both training and validation loss and accuracy. The continuous decrease in training loss, coupled with the consistent rise in training accuracy, signifies effective learning. Concurrently, the validation loss

showed a similar downward trend, confirming the model's ability to generalize. The early stopping mechanism based on validation loss further ensured the prevention of overfitting.

B. *Model Generalization*

The model's generalization capabilities were not only evaluated by its performance on both training and validation data, but also real world data not found in the Processed DAQUAR Dataset. The close alignment of training and validation metrics indicated the model's strong generalization ability. Moreover, the use of early stopping based on validation loss further enhanced the model's robustness, ensuring high performance on unseen data as seen in 3.

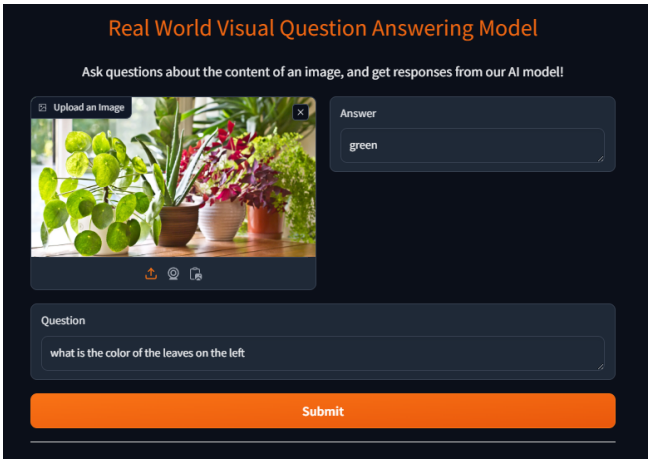

Fig. 3. Test on Unseen Data

C. *Similarity Scores*

Our model achieved an average SBERT similarity score of 0.67, indicative of a high degree of semantic correspondence between the predicted answer and the actual answer for any given question.

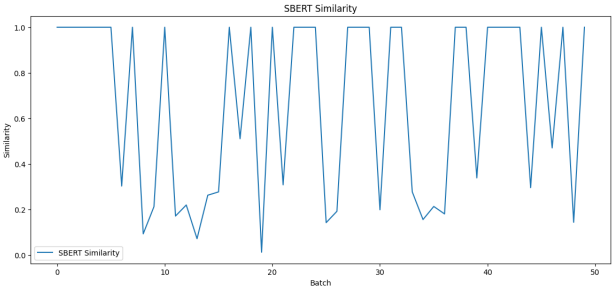

Fig. 4. SBERT Similarity Scores

Figure 4 shows the distribution of SBERT similarity scores across the validation set, with the majority of scores clustered in the higher range. This distribution underscores the model's capacity to produce semantically accurate answers. As can be seen from figure 5, the same answers indicate a complete semantic relationship between them by definition. Similarly, 6 shows that the model predicted shelves instead of drawer as the answer. However, as there is a high semantic relationship between them due to being pieces of furniture, the score is high. Lastly, we can see from 7 that the two objects possess non-existent semantic relationship with each other - hence bearing a low score.

Fig. 5. Maximum Similarity Score

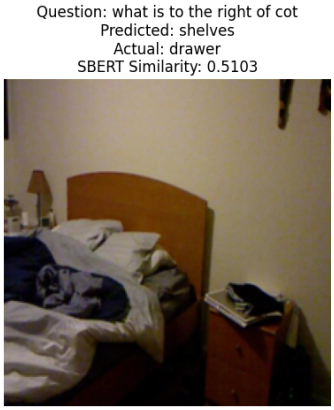

Fig. 6. High Similarity Score

Fig. 7. Low Similarity Score

VII. Discussion

The integration of ResNet50 and BERT enables our model to leverage state-of-the-art feature extraction techniques from both vision and language domains. However, the approach has a few limitations. The reliance on separate pre-trained models restricts the adaptability to specific scenarios with characteristics different from those of the training data of BERT and ResNet50. Additionally, the concatenation of features from both models, while effective, does not fully capture the complex interactions between visual and textual data - increasing the chance of random error.

VIII. Future Work

Future implementations can explore more sophisticated feature fusion techniques, such as attention mechanisms that dynamically weigh the contributions of image and text features. Expanding the dataset and incorporating diverse question types can also improve the model's robustness. Moreover, utilizing larger transformer models like GPT-4 or Vision Transformers (ViT) could further enhance the VQA capabilities.

IX. Conclusion

This report presents a novel approach to VQA in Real World Images by integrating ResNet50 and BERT. Our model achieves competitive performance on the Processed DAQUAR dataset, demonstrating the effectiveness of combining high-level visual and textual feature extractors - particularly ResNet50 and BERT. Future works can use this work as a foundation to further improve the model by applying it on broader datasets and question-answer pairs.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

[3] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.