Incorporating Vision Encoders into Retrieval Augmented Visual Question Answering

Tony Yan
supervised by Prof.Bill Byrne and Weizhe Lin
Department of Engineering, University of Cambridge

January 18, 2024

Abstract

This report discusses the background, methodology and the progress of integrating a vision encoder into the existing Retrieval Augmented Visual Question Answering (RAVQA) framework to enhance the performance on the Knowledge-based Visual Question Answering (KB-VQA) task. The modified model aims to improve image understanding and answer accuracy by embedding images directly into a vector space. Preliminary results show promise despite an initial slow learning rate, indicating potential improvements in precision and accuracy over the baseline model. Future work will focus on optimising the mapping layer and increasing the training time to further improve performance.

1 Introduction

The primary goal of this project is to solve the challenging task of Knowledge-base Visual Question Answering (KB-VQA), which is answering questions about an image by retrieving knowledge from a knowledge base, because the answer does not present in the image. An example is shown in Figure 1. The baseline model for this project is the model proposed by Retrieval Augmented Visual Question Answering (RAVQA)[1], which was the state-of-the-art model for KB-VQA. The understands the image by converting the image into a text representation including objects, corresponding captions and any characters detected. This is efficient, but there is a risk of losing information during the conversion. Therefore, the goal of this project is to introduce a vision encoder to the model to encode the image into a vector in an embedding space so that a better performance is achieved.

Figure 1: Question: What are the health benefits of this vegetable? Answer: Fiber

2 Methodology

Figure 2 shows the overview of the modified model with the same vision encoder added to two different places by connecting to different mapping layer. The black text blocks represent the original RAVQA model, and the blue text blocks represent the modified parts. Both the baseline model and the modified model are three-stage models:

- Image to text including objects, captions, and characters (not vision encoder).
- Document Retriever(DPR): retrieve the top k documents by similarity search.
- Answer Generator(RAG): generate the answer candidates from the retrieved documents, question, and the converted text.

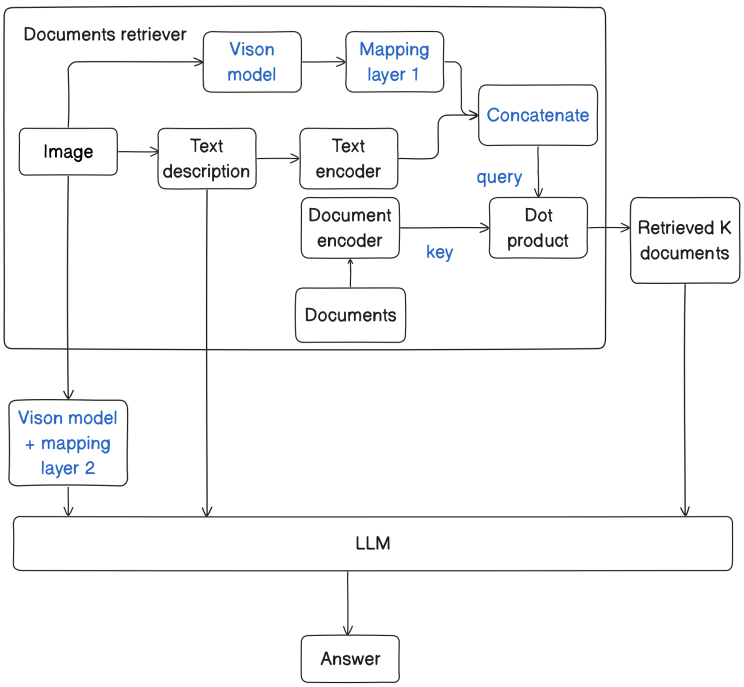

Figure 2: Overview of modified RAVQA model. The black text blocks represent the original RAVQA model, the blue text blocks represent the addition parts. Same vision encoder with different mapping layers are added to the model.

Informations loses during the conversion from image to text representation. For example, the spatial information. Therefore, one vision encoder is introduced twice with different mapping network in the document retriever and the answer generator respectively so that the model can fully understand the image, improving the precision of the retrieved passages and the generated answers.

2.1 Vision Encoder

A vision encoder is a model (usually a CNN or a transformer architecture) that encodes an image into a vector in an embedding space. The vision encoder introduced is a pretrained

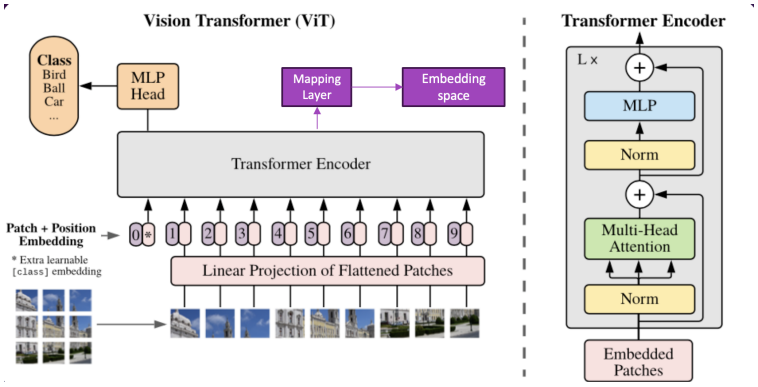

Figure 3: Vision encoder with mapping layer (purple block)

CLIP-vit-base-patch32 model from OpenAI [2]. In order to join the vision encoder with the baseline model, a mapping layer is added to the embedding output of the vision encoder to match the dimension of the output, and map the original embedding output to a new embedding space (as shown in Figure 3) so that the baseline model can better understand the image during training. There are many types of mapping layers, such as a fully connected network, and a Q-former architecture [3]. In this report, two single linear layers without an activation function,the simplest mapping layer, are used for rapid prototyping and examining the effectiveness of the chosen vision encoder.

2.2 Document Retriever

With the question encoded as $F_q(x)$, and the image encoded as $F_i(i)$, where x is the converted text and i is the image, the query vector is defined as:

$$q = \frac{F_q(x) + F_i(i)}{2} \tag{1}$$

The key vector is defined as $k = F_d(z)$, where $F_d(z)$ is the encoded documents or passages, and z is the converted text of the documents. The similarity score is calculated as:

$$s_d(x, i, z) = q^T(x, i)k_d(z) \tag{2}$$

and the probability of choosing the document is calculated as a softmax:

$$p_d(x, i, d) = \frac{exp(s_d)}{\sum_{d'} exp(s_{d'})} \tag{3}$$

The selected k documents are the top k documents with the highest probability.

2.3 Answer Generator

Before training the answer generator, a FAISS index in created for fast dynamic document retrieval. The answer generator in this model is the T5ForConditionalGenerator [4] which only allows one input types (either encoded token ids or embeddings). Therefore, all

the inputting information including the question, the captions and the retrieved documents must be converted to embeddings before feeding.

For a given document z_k, the answer is generated as:

$$y_k = \arg\max_y p_a(y|z_k, x, i) \tag{4}$$

Where y_k is the answer generated from the document z_k, and $p_a(y|z_k, x, i)$ is the probability of an answer given the document, text and image.

One answer is generated for each document, and the final answer is selected based on the probability of the answer and the probability of the document:

$$\hat{y}, \hat{z} = \arg\max_{y, z_k} p_a(y|z_k, x, i) p_d(z_k|x, i) \tag{5}$$

where z_k is the selected document from the top k documents, and $p_a(y|z_k, x, i)$ is the probability of an answer given the document, text and image.

2.4 Training Method

The model is trained in two stages. The first stage is to train the document retriever with the cross entropy loss between the predicted probability and the ground truth. The second stage is to train the answer generator. For each document, the answer generator produces an answer with the highest probability and the loss is calculated as the cross entropy loss between the gold answer assigned to that document and the generated answer.

The vision encoder [2] model is pretrained on a large dataset, so the vision encoder is frozen during the training process and only the mapping layers are the trainable part from the vision encoder so that the model keeps its image understanding and is not being ruined by the training process.

3 Progress

- Set up model environment and learnt how to use HPC. The model environment is from 2021, so the latest packages are not compatible. It is difficult to train a large model in a local machine, so the HPC is necessary.
- Learnt the RAVQA model [1] and the CLIP-vit-base-patch32 model [2].
- Reviewed literature about recent large multi-modal models to learn their architectures and training strategies.

3.1 DPR

Due to the addition of the vision encoder, the training time has increased by 30% compared to the original model. Therefore, fewer epochs are used to train it.

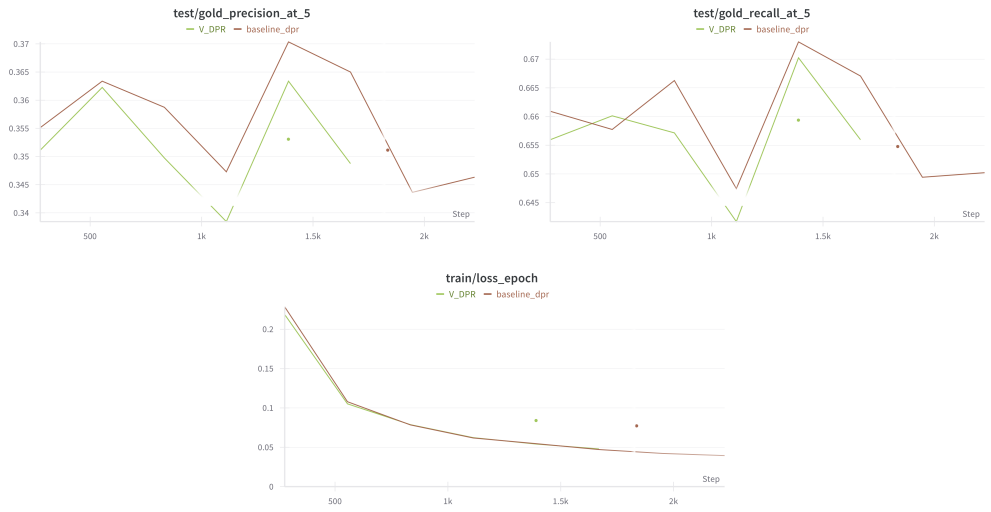

Figure 4: Results for training DPR. Precision is the percentage of the top k (1 or 5 here) documents that contain the answer. Recall is defined as: Recall $= \frac{\text{TP}}{\text{TP+FN}}$ where True Positives (TP) are the number of positive documents correctly identified by the model. False Negatives (FN) are the number of positive documents that the model incorrectly classified as negative.

Figure 4 shows the training loss and the evaluation results for the DPR model. The training loss of the modified model descended slower than the baseline model. The evaluation results indicate that performance of the modified model is worse than the baseline DPR model in DPR training process. The reason is that the mapping layer is too simple so the image embeddings is not properly projected.

3.2 RAG

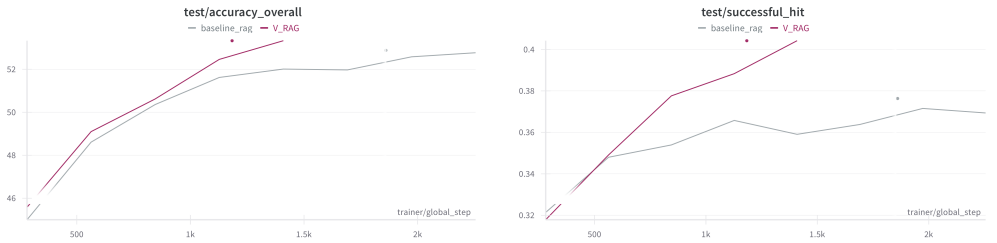

Figure 5: Results for training RAG. Overall test accuracy (left) and successful test hit rate (right). Hit rate is the percentage of the test samples that the model successfully selected the correct document.

The RAG model with a vision encoder has been implemented and trained under the same dataset and training strategy as the original RAVQA model [1]. This time, the flan-T5 conditional generation model does not support ids and embeddings input at the same time,

so the input ids are converted to embeddings before feeding. The dimension of the text embeddigs are (batch size, seqence length, embedding dimension = 1024). The dimension of the image embeddings are (batch size,1, embedding dimension = 1024). The final embeddings inputs are the concatenation of the text embeddings and the image embeddings.

The results for training the RAG model are shown in Figure 5. The testing results of the modified model is slightly better than the baseline model in general, and it keeps ascending fast. The hit rate increases much faster than the baseline model, indicating that the better image understanding helps the model to select the correct document and generate the correct answer.

4 Conclusion and Future Work

The results show that the modified model learns slower than the baseline model during the DPR training stage, but both the answer generator and the document retriever learn much faster during the jointly training stage.The overall performance has outperformed the baseline model, indicating that the vision encoder has improved the performance of the model.

The next step is to improve the mapping layer so that the image s can be properly projected, more layers of fully connected networks and Q-former [3] architecture will be tested. In addition, the training time will be increased to see if the model can achieve better performance. Furthermore, a model without image to text conversion will be implemented to see how useful the image to text conversion is for both the document retriever and the answer generator.

As described in section 3.2, the image fed into the answer generator is a tensor with sequence length one, while the sequence length of the text is usually hundreds. Therefore, further work will be done to transform the image embeddings to a tensor with longer sequence length or feed the model with encoded image and the encoded region of interests for better image understandings.

References

[1] Weizhe Lin and Bill Byrne. Retrieval augmented visual question answering with outside knowledge, 2022.

[2] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

[3] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023.

[4] Thomas Wolf et al. Huggingface's transformers: State-of-the-art natural language processing, 2019. Accessed: 2024-01-16.