# CLIQQA: Image-Based Question Answering Using CLIP Embeddings

**Pendo Abbo**
Columbia University
pa2451@columbia.edu

## Abstract

This work leverages pre-trained CLIP and DistilBERT models to train a network that performs image-based question answering on the Toronto COCOQA dataset. Image embeddings are derived from a pre-trained CLIP vision encoder and concatenated to the word embedding sequence of a question. A DistilBERT language model uses the resulting representation as input to generate an answer to the image-based question. Our results show significant improvements in test accuracy when the CLIP embeddings are first passed through a small MLP network to obtain the final image embeddings. We achieve further improvements in the test accuracy after fine-tuning the DistilBERT model for the task.

## 1 Introduction

Image-based question answering is a multi-modal task that combines image understanding with natural language understanding by requiring the model to learn the semantic information represented in an input image and generate the correct answer to a text-based question about the contents of the image. This extends beyond traditional computer vision tasks of image classification and object identification since the model must learn more granular characteristics of the image in order to answer a question about the image. This also goes beyond text-based question-answering tasks in natural language processing, such as SQuAD (Rajpurkar et al., 2016), which have an accompanying text-based document which the model must use as contextual knowledge to answer the provided question. For image-based question answering, the supporting document is an image. Therefore, our challenge is to obtain a representation of the image that is both rich in semantic encoding and closely related to a text-based representation of the image.

In order to address this challenge, we propose using the pre-trained CLIP (Radford et al., 2021) model to obtain embedding representations of the images. CLIP is a multi-modal vision and language model pre-trained on the task of matching images to their captions. Our expectation is that this pre-training objective requires the CLIP vision encoder to learn a representation of an image that encodes its semantic contents in order to successfully match the image to its caption. As such, we extract image representations from the CLIP vision encoder.

In order to generate a text answer to the question, we require a language model. We've selected the pre-trained DistilBERT (Sanh et al., 2019) model given its success in a variety of natural language understanding tasks. However, the model typically receives an input sequence of word embeddings, and although we expect the CLIP image embeddings to contain semantic knowledge of an input image, these embeddings do not inherently lie in the same latent space as word embeddings. To remedy the issue, we propose an intermediate mapping network to transform the CLIP embeddings into vectors that ideally lie in the same latent space as word embeddings. The output of this network will be treated similarly to word embeddings of a supporting document in the SQuAD task, and concatenated to the sequence of word embeddings of the input question. Our expectation is that such a mapping network will yield a representation of the input image that is more suitable to our language model.

Our model implementation can be found at https://github.com/pmabbo13/imageqa.

## 2 Related Work

Work done by (Ren et al., 2015) introduced the Toronto COCOQA dataset and experimented with a CNN+LSTM multi-modal network to train their

dataset on the task of image-based question answering. They operated under the stipulation that all answers must be one-worded, and therefore treated the task as a classification problem such that the model must choose the most probable word in the vocabulary as the answer to the input question. This assumption was made to avoid ambiguities around evaluating the accuracy of sentence-level answers. Though metrics such as ROUGE (Lin, 2004) and BERTScore (Zhang et al., 2020) allow for comparison between system generated and reference sentences, they do not perfectly capture the task accuracy. We therefore take the same approach as (Ren et al., 2015) and treat the problem as a classification task, only considering one-worded answers.

To the best of our knowledge, our work is novel in applying our proposed architecture to the task of image-based question-answering on this dataset. Our architecture takes inspiration from the ClipCap model (Mokady et al., 2021), which also used a pre-trained CLIP vision encoder and trained an intermediate mapping network between its vision and language models. Our work deviates from the ClipCap model in that ClipCap used an autoregressive language model to generate sentence-level image captions. They also experimented with using both an MLP and Transformer mapping network, whereas we only use an MLP mapping network.

## 3 Data

We use the Toronto COCOQA dataset (Ren et al., 2015), which was curated for image-based question answering. The dataset covers four types of questions:

1. what object is present in an image
2. the color of an object in an image
3. the number of a certain object in an image
4. the location of an object in an image

All answers are one word and the images are originally sourced from the COCO dataset (Lin et al., 2014). In total, the dataset uses 69,172 distinct images, for which there are 78,736 distinct (image, question) pairs in the training set, and 38,948 distinct (image, question) pairs in the test set. We randomly select 10% of the training data to be used as our validation set. Examples from the dataset are shown in Figures 1 and 2.

Figure 1: **[Question]** What is displayed in the clear crystal vase? **[Answer]** Flowers

Figure 2: **[Question]** How many people are there in an office posing for a picture? **[Answer]** Six

## 4 Methods

Our model architecture leverages pre-trained versions of CLIP and DistilBERT as our vision and language models respectively. We first pass images through the CLIP vision encoder and extract embeddings from the final hidden layer to be used as our initial image embeddings. We've chosen the CLIP model because it was pre-trained on the task of matching images to their captions. Our expectation is that image embeddings derived from this model will encode the semantic information of an input image. Such an encoding should prove useful for image-based question answering since our model will need to have some language understanding of the image in order to correctly answer a question that is based on what is observed in the image.

Following the vision encoder is an intermediate MLP network that is used to project the CLIP embedding to the textual latent space. We consider the output of this intermediate network to be our final image embedding, which we then concatenate to the sequence of word embeddings of the accompanying question (as shown in Figure 3).


Figure 3: Concatenation of question and image embeddings. This is used as the input to the DistilBERT model.

We use the resulting vector as the input to our pre-trained DistilBERT language model. DistilBERT is a version of the seminal BERT model (Devlin et al., 2018) which has been shown to perform well on natural language understanding tasks including text-based question answering. DistilBERT uses a knowledge distillation technique (Hinton et al., 2015) to reduce the number of model parameters by 40% while preserving 97% of BERT's performance on a variety of natural language understanding tasks. We therefore expect that using DistilBERT as opposed to BERT will speed up the training and inference time without severely compromising the task performance.

Since we have formulated our problem as a classification task over the vocabulary, we add a linear layer to project the final hidden layer of the DistilBERT model to a vector that is the size of our vocabulary. We then use a softmax to select the most probable word in the vocabulary as the predicted answer to the input question. A diagram of our architecture is shown in Figure 4

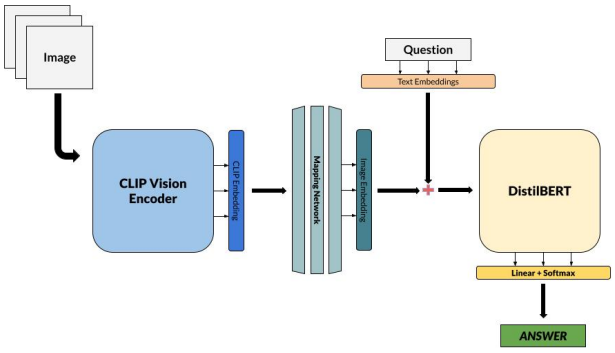


Figure 4: Model architecture

## 5 Experiments

Since the pre-trained CLIP model has shown success in zero-shot settings for transfer learning tasks, we freeze all of its parameters and do not update its weights during training. We then train four versions of our model architecture by varying the size of the mapping network, whether or not we freeze the pre-trained DistilBERT model weights, and the number of epochs we train for. All experiments are trained using a cross-entropy loss between the predicted word and the correct answer. We use an Adam optimizer for training with $betas = (0.9, 0.999)$, $eps$ =1e-8, and $weight\_decay = 0$.

Our first variant freezes the DistilBERT weights and contains no intermediate mapping network. This means the CLIP embeddings are directly used as the image embeddings that are concatenated to the word embeddings of the question. We train for one epoch with a learning rate of 1e-4 to learn the weights of the final linear layer that follows the DistilBERT model. This variant is used as our baseline.

The second variant keeps the text model frozen and introduces an MLP mapping network that consists of two linear layers with hidden dimensions of size 1,024 and ReLU activation functions. We use the output of this small network as the image embeddings which we concatenate to the word embeddings of the question. We train the entire network for one epoch with a learning rate of 1e-4. The results of this model indicate the effects of introducing the mapping network when compared to results from the baseline.

Our third variant starts with the weights of the second variant in order to use the earlier training as warmup for the randomly initialized weights that we've introduced in our network. We then unfreeze the text model and train for one more epoch with a smaller learning rate of 1e-5. The results of this model indicate the effects of fine-tuning the DistilBERT model on the task when compared to the results of variant two.

Our last variant trains the third variant for an additional epoch using the same learning rate to see how training for longer will affect the task performance.

## 6 Results

Our results are shown in Table 1. The baseline model achieves an accuracy of 28.8%, indicating that using the CLIP and DistilBERT models in zero-shot format without an intermediate mapping network yields poor performance on the task. Upon adding our small MLP mapping network, we are able to increase the accuracy to 51.3%. This indicates that our mapping network is able to succesfully transform the CLIP embeddings into representations that are more useful to our language model. Finally, we gain additional improvements in test

accuracy by updating the language model weights as well. Under such conditions, training for one and two more epochs increases the model accuracy to 55.0% and 62.5% respectively.

| | Num. Mapping Layers | Frozen Text Model | Training Epochs | Test Accuracy |
|---|---|---|---|---|
| **Version 1** | 0 | True | 1 | 28.8% |
| **Version 2** | 2 | True | 1 | 51.3% |
| **Version 3** | 2 | False | 2 | 55.0% |
| **Version 4** | 2 | False | 3 | 62.5% |

Table 1: Test Results

## 7 Conclusion

Our work shows that we can achieve 62% accuracy on the task of image-based question answering using the Toronto COCOQA dataset by leveraging pre-trained CLIP and DistilBERT models and training an small intermediate mapping network between the two. Future work may experiment with using a more sophisticated mapping network, either by including more linear layers or adopting an entirely different architecture such as a transformer. We may also consider using a larger language model, such as BERT, to see if we can further improve on task performance. Finally, though there may be challenges with evaluation metrics, future work may consider applying this general architecture for image-based questions with multi-word and sentence-level answers.

## References


Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. 2014. Microsoft coco: Common objects in context. arXiv.

Ron Mokady, Amir Hertz, and Amit H. Bermano. 2021. Clipcap: Clip prefix for image captioning.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. arXiv.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. 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, Austin, Texas. Association for Computational Linguistics.

Mengye Ren, Ryan Kiros, and Richard Zemel. 2015. Exploring models and data for image question answering. In *NIPS*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.