Vision is Language: Visual Understanding via LLM

Xiangyu Liu
Stanford
jxiangyu@stanford.edu

Abstract

Many vision tasks can benefit significantly by leveraging language. Some tasks, such as Visual Question Answering (VQA) for open-ended questions, require the vision model to know language. In this project, I'll explore how to use language models to improve the open-ended VQA accuracy.

1. Introduction and Problem Statement

When we hear "sky is blue", we can immediately picture a beautiful blue sky above with a few white clouds in our mind. Similarly, when we see a dog chasing after a ball, we can effortlessly describe which is doing what. Hence, intuitively, there is a large overlap between vision understanding and language understanding.

As the Large-Language Model has exhibited promising capability recently, research to leverage LLM to improve image understanding has witnessed a rapid advancement. One fundamental capability of visual understanding is to perform visual question answering (VQA). Many different approaches [such as (4), (1)] that leverage vision encoder and LLm for VQA tasks have been explored to improve the overall accuracy of VQA tasks.

However, the open-ended VQA remains a very challenging task, because it not only involves many traditional vision tasks, such as object segmentation, but also requires the natural language understandings. More importantly, it needs to be able to identify the connection between vision tasks and the language in order to answer vision related questions.

In this paper, I plan to use the VQA and CoCo datasets to train and evaluate the accuracy of the neural network for VQA performance, with the focus on the open-ended questions instead of the True/False questions or multiple-choice questions.

2. Methods

2.1. Architecture

To properly answer the open-ended questions, the model needs to understand language and how language and vision interact with each other. Hence, it is important to leverage the image and its captions in order to ensure the text-alignment in the feature space of the image. My high level model architecture is shown below.

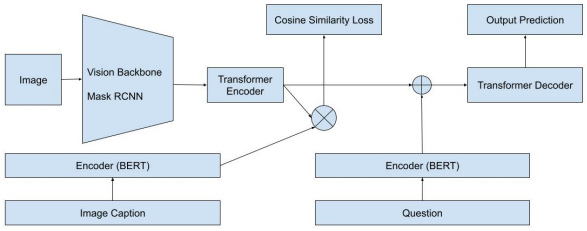

Figure 1. Model Architecture

2.2. Vision Backbone

Similar to (3), the images are first transformed into feature space. However, just image embedding is not sufficient because many objects in the images can have spatial relationship. Therefore, extracting the spatial related features is equally important. Therefore, I plan to adopt the Mask R-CNN (2) as the backbone to extract image features as well as their spatial information.

2.3. Caption Embeddings, Feature Projection, and Similarity Loss

To understand the relationship between the image features and the natural language, all the captions of a given image are encoded into feature spaces using an text encoder.

The image and spatial features extracted by the Vision Backbone are then projected onto the text embedding space via a transformer. Both embeddings will be used to compute a cosine similarity loss, in order to make sure the projected image embeddings are similar to the text embeddings.

2.4. Open-ended VQA

Before the model can perform the QA tasks, the question is also encoded into embedding space using the same frozen encoder that was used to encode the image caption. This is to make sure the consistency in the embedding space.

The projected image embedding from the earlier step will be concatenated with the question embedding to form the context vector. A transformer decoder is then deployed to output the final answer.

2.5. Learnable parameters and Loss function

The Vision Backbone and the Text Encoders are both frozen. The only trainable parameters are the Transformer Encoder that projects the image features into text embedding space and the transformer decoder that output the final answer prediction.

The total loss here is the sum of the cross-entropy loss for the final answer prediction and the cosine similarity loss.

3. Current Progress and Intermediate Results

So far, I've prepared the data, visual backbone, as well as the encoder. I've also designed the architecture and integrated the CoCo data from FiftyOne into Pytorch datasets.

Next step would be to complete the implementation of the architecture and start training.

References

[1] X. Chen, X. Wang, S. Changpinyo, A. Piergiovanni, P. Padlewski, D. Salz, S. Goodman, A. Grycner, B. Mustafa, L. Beyer, et al. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*, 2022.

[2] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[3] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[4] W. Wang, H. Bao, L. Dong, J. Bjorck, Z. Peng, Q. Liu, K. Aggarwal, O. K. Mohammed, S. Singhal, S. Som, et al. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *arXiv preprint arXiv:2208.10442*, 2022.