Examining Vision-Language Models

Conrad Li, Rebecca Du, Anish Parmar

March 13, 2025

1 Introduction

Vision-Language Models (VLMs) integrate visual and textual modalities to generate textual descriptions of images or answer questions based on image content. This report outlines the implementation and evaluation of a generative ML system for interpreting image data.

2 Basic Requirements

2.1 System Implementation

Implement a basic encoder-decoder system for image captioning task, consisting of:

- **Vision Encoder:** Pre-trained ViT (e.g. CLIP) to encode images as feature representations.
- **Language Decoder:** Lightweight pre-trained LLM (e.g. DistilGPT-2) to generate textual outputs.

2.2 Model Tuning and Evaluation

Fine-tune language decoder on an image captioning dataset sample (e.g. COCO), and evaluate performance using standard metrics such as:

- BLEU
- CIDEr
- SPICE

2.3 Text Decoding Strategies

Implement a text decoding strategy to improve generated text outputs:

- Beam search
- Others

3 Potential Optional Requirement(s)

*Final implemented option(s) may be limited due to compute capacity

3.1 Visual Question Answering (VQA)

Expand system task to handle simple VQA tasks by incorporating:

- **Input Language Encoder:** Lightweight pre-trained LLM (e.g. DistilBERT) to encode textprompt inputs.
- **Feature Fusion Module:** Simple (e.g. Concatenation) or sophisticated (e.g. cross-attention mechanism) to merge vision and text encoder embeddings as LLM input.

3.2 Prompt Engineering

Augment LLM input with hidden, learnable prompts from vision encoder output to enhance text generation results.

3.3 Vision Encoder Fine-tuning

Implement fine-tuning on the vision encoder:

- Freeze LLM decoder weights for supervised learning on encoder.
- Fine-tune vision encoder separately with self-supervised learning.

4 Heilmeier Catachism

4.1 What are you trying to do?

We want to build a generative ML system that can interpret the information in pictures.

4.2 How is it done today?

A ViT processes image patches into text embeddings using self-attention. These embeddings are mapped to the embedding space of an LLM so that they can be used equivalently as a normal text prompt input. The LLM itself also uses self-attention to generate output tokens.

4.3 Your approach and why do you think it will be successful?

We will develop the VLM using a pre-trained ViT encoder as inputs for a pre-trained, lightweight LLM decoder. A projection layer will need to be included to map the ViT outputs to the embedding space of the chosen LLM. For fine-tuning the base model, we will freeze the weights of the encoder and use an open-source supervised dataset to train the LLM.

We believe this project can be successful because transformer models have demonstrated state-of-the-art performance in both vision and language tasks. The pre-trained LLMs therefore already likely provide a reasonable starting point for evaluation. Furthermore, there are good open-source datasets for tuning image captioning (e.g. COCO) and VQA tasks (VQA2).

4.4 What are the risks?

Potential risks are:

- Compute limitations restricting us to smaller models may not be able to effectively perform desired task.
- Multi-component architecture for VLMs increases tuning complexity.

4.5 How long will it take?

The project is planned for a 5-week duration.

4.6 What are the final "exams" to check for success?

Success is determined by:

- Image captioning performance using BLEU scores.
- Empirical testing on unseen images and questions.

5 Conclusion

This proposal aims to develop a VLM system for image interpretation and text generation. Success will be evaluated through quantitative benchmarks and qualitative assessments.