Combining Pre-Trained Unimodal Models with a Single Fully-Connected Layer

Quinn Tucker
Rochester Institute of Technology
Rochester, NY, USA
qt2393@rit.edu

***Abstract*—Large vision-language models are expensive to train from scratch. However, recent work has demonstrated that pre-trained unimodal models can be combined by connecting them with a learned "adapter". In this project I experiment with a single fully-connected layer as the adapter and show that the resulting system has nontrivial multimodal capabilities.**

I. Introduction

The recent explosion of large language models shows great promise for the creation of general-purpose assistants or for automating tasks on unstructured input. However, autoregressive language models trained on text are not natively able to process input from other modalities, such as images. There is much utility to be gained by enabling LLMs to accept visual input along with textual prompts.

Unfortunately, training a large multimodal model from scratch is expensive. Previous work has therefore investigated the possibility of leveraging strong unimodal models to construct multimodal systems, via techniques such as:

- Adding an vision module to a language model, then fine-tuning the augmented LM on multimodal data.
- Training only an image encoder that feeds into a *frozen* text-only LM [1].
- Training an "adapter" module that connects a *frozen* image encoder to a *frozen* text-only LM [2].

The third approach is the subject of my project. Prior work (such as BLIP-2 [2]) uses a transformer-based adapter module to map the image encoder's output to the language model's input. But the lack of ablation experiments in [2] raises the following question: **how much can we simplify the adapter module between the two modalities?** In this project, I test the effectiveness of arguably the simplest possible adapter design: a single fully-connected layer.

II. Methodology and Experimental Design

A. Model Architecture

Figure 1 shows the architecture of the system used in my experiments. An input image is first processed by a pre-trained image encoder network. The resulting image embedding is then passed through a small **adapter module**, which maps it to one or more "pseudo-tokens" in the language model's token embedding space. Finally, these adapted embeddings are inserted into the sequence of token embeddings in the language model's prompt.

The adapter module consists of a single fully-connected layer. Its parameters are a matrix $W \in \mathbb{R}^{d_\text{I} \times (k d_\text{L})}$ and a bias vector $b \in \mathbb{R}^{k d_\text{L}}$, where d_I is the dimensionality of the image encoder's output embeddings, d_L is the dimensionality of the language model's token embeddings, and k is the number of pseudo-tokens that the adapter produces. Given an image embedding $e_\text{I} \in \mathbb{R}^{d_\text{I}}$, the adapter module therefore performs an affine transformation to compute the adapted embeddings $e_\text{L} \in \mathbb{R}^{k d_\text{L}}$ (Equation 1).

$$e_\text{L} = e_\text{I}^\top W + b \tag{1}$$

In all my experiments, I use CLIP ViT-L [3] as the image encoder (for which $d_\text{I} = 1024$) and Mistral 7B Instruct [4] as the language model (for which $d_\text{L} = 4096$). An adapter module that outputs $k = 1$ token therefore has $1 \cdot 4096 \cdot (1024+1) = 4{,}198{,}400$ parameters, which is ~100× and ~1700× smaller than the CLIP and Mistral models, respectively.

B. Training

Similar to the approach in [2], I train the parameters of the adapter module with an image captioning task using the CC12M dataset of image captions curated for image-text pre-training [5]. Specifically, given an (image, caption) pair, the system is trained to maximize the log-likelihood of generating the sequence of caption tokens, following the adapted image embedding as a "prompt". Although this involves back-propagating gradients through the language model, both the image encoder and language model are kept completely frozen; only the parameters of the adapter module are updated.

I trained three adapters that output $k = 1, 2, 5$ pseudo-tokens, respectively. Due to time and resource constraints, each model was only trained on a subset of the CC12M dataset (200k instances for the $k = 1, 2$ models and 80k instances for the $k = 5$ model) – although they appear to have converged before that point. For other hyperparameters and implementation details, the reader is referred to the `train.py` script in my submitted code.

C. Evaluation

I evaluate the trained system on the task of zero-shot visual question answering, in which I present the system with an image and a question about the image,

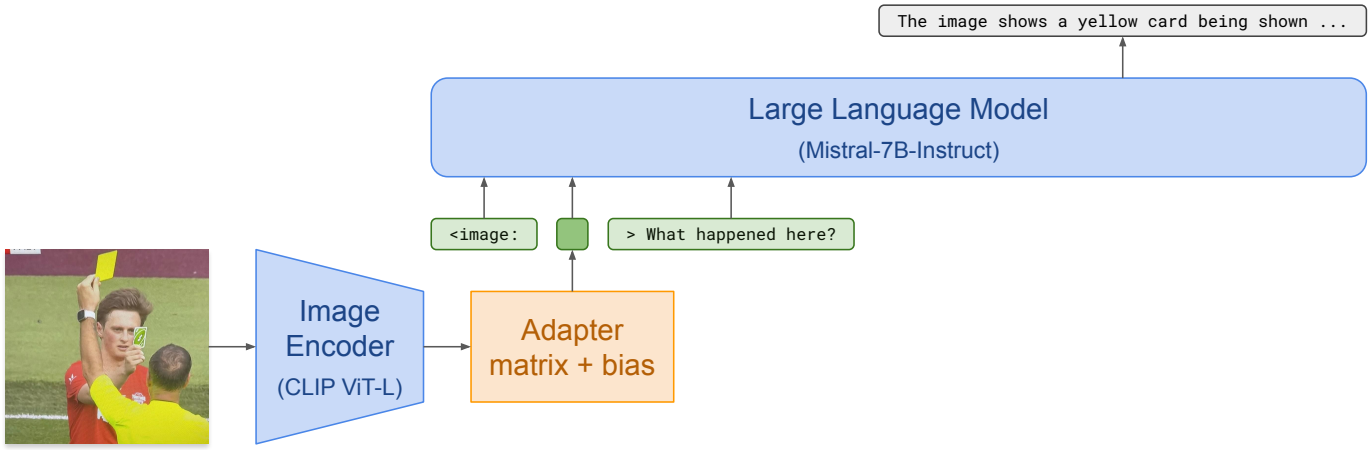

Figure 1. The architecture of the multimodal LLM system in my experiments. The unimodal image encoder and language model (blue) are both kept frozen from pre-trained checkpoints; only the parameters of the adapter module (orange) are trained on multimodal data.

and the model is expected to answer the question. I use the prompt format `[INST] <image:`e_I`> Given this image:` q `[/INST]` (including Mistral's instruction framing), where e_I is the sequence of pseudo-tokens from the adapter and q is the text of the question. I use simple greedy decoding to generate the model's response.

Due to time constraints and Mistral's preference for long-form answers, I feed the model $n = 50$ random (image, question) pairs from the validation set of the VQAv2 [6] visual question answering benchmark and manually judge the accuracy of the model's open-ended responses.

Additionally, I investigate the interpretability of the adapted pseudo-token embeddings by comparing them to embeddings of tokens in the language model's vocabulary.

III. Results

All three adapters ($k = 1, 2, 5$) attained nearly identical training losses and I (informally) cannot discern any consistent difference between the quality of their outputs. For some prompts, the different models generate identical text. In hindsight this is consistent with the observation that a full-rank adapter matrix can in theory transfer all of the information in the 1024-dimensional CLIP embedding into a single 4096-dimensional Mistral embedding. Indeed, inspecting its eigenvalues reveals that the learned single-token matrix is very nearly full-rank. So, due to time limitations and also for brevity, all of the following results are from the trained adapter that produces $k = 1$ pseudo-token per image. A more thorough investigation of the effect of pseudo-token count is left to future work.

A. Responses to VQAv2 Questions

Table I gives a quantitative comparison between my system, Frozen, and BLIP-2 on the task of zero-shot visual question answering. My system answered 22 out of the 50 questions correctly (44%). The most comparably-sized BLIP-2 model falls within my model's confidence interval, but my system does not appear to match the performance of the best BLIP-2 model. However, my system compares favorably to Frozen and has dramatically fewer parameters trained on multimodal data than any of the other models.

In most cases, the system is able to correctly identify *high-level* semantic attributes of the scene (urban street, cat, vintage black-and-white photograph, bikes) as well as subsequently connect those attributes to the world knowledge baked into the language model. These capabilities are occasionally quite impressive and fine-grained given the small visual information bottleneck, such as identifying the tennis player Roger Federer or the words "Feliz Navidad" stitched on an otherwise-generic teddy bear.

However, the model frequently makes obvious blunders. I noticed several categories of failures while evaluating my system's responses, outlined in Table II. Many of the failures highlight a difficulty with specifics: the model struggles to count objects, report non-stereotypical colors, and describe attributes of objects that are not the center of attention in the image's composition. In other cases, the model hallucinates objects that are not present in the image (such as clouds in a clear blue sky or nonexistent objects on a desk) or confuses an object for something related (e.g. referring to sheep as goats or llamas).

B. Interpretability of the Adapted Image Embeddings

Table III shows the 15 tokens from the language model's vocabulary that have the greatest cosine similarity with the adapted pseudo-token embedding for the image of the soccer player in the lower-left corner of Figure 1. A few of these tokens make some sense in the context of the picture: "Player"; "Championship" and "чемпиона" (Russian for "champion"); and a few names that could conceivably be athletes. The rest, in particular those at the very top of the list, are inscrutable, with no obvious connection to the image.

A similar pattern emerges for other natural images: occasionally a few tokens that make sense (like "sand" in English and Chinese for an image of a desert), but largely gibberish including software-related terms ("ldots", "TagHelpers"), punctuation, and various Unicode symbols.

Model	# Trainable Params	# Total Params	VQAv2 Val Accuracy	95% conf. interval	n
Frozen [1]	40M	7.1B	29.6	–	–
BLIP-2 ViT-g $\text{OPT}_{6.7B}$ [2]	108M	7.8B	54.3	–	–
BLIP-2 ViT-g FlanT5_{XXL} [2]	108M	12.1B	**65.2**	–	–
Mine	4.2M	7.7B	44.0*	[30.0, 58.7]	50

Table I. Zero-shot visual question answering accuracy on the VQAv2 validation set. *My system was manually evaluated on a small subset of the data, as indicated.

Failure Mode	Example
Model doesn't give a definite answer	"It is difficult to determine if the water is turned on based on the image alone."
Model reports inaccurate object count	The model says there are 12 donuts in an image showing only 2.
Extra objects hallucinated	The model describes extra objects on a desk that are not present in the image.
Confusing a related type of object	The model says there are llamas in an image containing sheep.
Model reports inaccurate pose/orientation	The model says a boy has his arms raised, when his arms are actually down.
Model reports inaccurate object color	The model says a helmet is white, when it is actually black.

Table II. A non-exhaustive categorization of observed failure modes in my system when performing zero-shot VQA.

Cos Sim.	Token
0.0661	$:
0.0590	)>
0.0585	uple
0.0581	_Fort
0.0579	_чемпиона
0.0577	_Hub
0.0577	Player
0.0572	业
0.0571	_Stefan
0.0568	_Constitution
0.0568	_creativity
0.0563	_Morgan
0.0559	_bewerken
0.0557	ldots
0.0553	_Championship

Table III. The LM vocabulary tokens that are most similar to the adapted pseudo-token embedding for the image at the left side of Figure 1. Mistral 7B's vocabulary has ~32,000 tokens in total.

The distribution of cosine similarities in Table III is also representative of all of the images I investigated; all similarities between an adapted image token and a vocabulary token were less than 0.08, indicating that the image tokens are largely orthogonal to the language model's vocabulary. This is comparable to the similarity between a random (Gaussian) vector in the embedding space and its closest vocabulary embedding, which is about 0.0637 on average.

Furthermore, replacing the image's pseudo-token with its most similar vocabulary item eliminates the model's ability to summarize the image. Thus the image embeddings are not simply being mapped to normal vocabulary tokens.

IV. Discussion and Conclusion

The authors of [2] report that BLIP-2 achieves ~50-60% accuracy on VQAv2 with comparably-sized vision and language models. Despite the many variables (different training data, model architectures, etc.) confounding the comparison between that and my work, it is clear that BLIP-2 is able to answer fine-grained questions about the contents of images with considerably higher accuracy.

One potential explanation may stem from the fact that the "Q-Former" adapter module in BLIP-2 has access to all of the individual patch embeddings from the ViT, while my adapter only uses a single pooled embedding. Thus my system is able to capture the overall "gist" of a scene, without being able to accurately report details of individual objects.

Regardless, it is notable that a wide range of visual-linguistic concepts are able to be communicated through just a single (pseudo-)token in the language model's input. And the fact that CLIP's output and Mistral's input can be connected with a single affine transformation suggests some common linear structure between the two vector spaces. On the other hand, the fact that the Mistral LLM can be "hacked" to interpret a single token as a description of an entire image, along with the apparently alien nature of the pseudo-token embeddings themselves, highlights the sadly familiar black-box nature of deep learning systems.

All in all, this project demonstrates a simple way to inject non-textual information into LLM prompts, while underscoring the difficulty of mechanistically understanding the inner workings of contemporary NLP models.

References

[1] M. Tsimpoukelli, J. L. Menick, S. Cabi, S. M. A. Eslami, O. Vinyals, and F. Hill, "Multimodal few-shot learning with frozen language models," in *Advances in Neural Information Processing Systems*, 2021.

[2] J. Li, D. Li, S. Savarese, and S. Hoi, "BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," *arXiv preprint arXiv:2301.12597*, 2023.

[3] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *Proceedings of the 38th International Conference on Machine Learning*, 2021.

[4] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed, "Mistral 7B," *arXiv preprint arXiv:2310.06825*, 2023.

[5] S. Changpinyo, P. Sharma, N. Ding, and R. Soricut, "Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts," in *Conference on Computer Vision and Pattern Recognition*, 2021.

[6] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, "Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering," in *Conference on Computer Vision and Pattern Recognition*, 2017.