MINI PROJECT 1: MULTIMODAL LEARNING

TECHNICAL REPORT

Suraj Anand
Brown University
suraj_anand@brown.edu

ABSTRACT

In this report, I describe experiments to build an image captioning model. I first performed zero-shot image classification with CLIP on CIFAR-10, achieving about 48.5% accuracy. Then, I trained a linear classifier to probe CLIP embeddings to determine whether this was a more effective method of image classification. Finally, I compared the image-captioning abilities of stitching together CLIP and GPT-3 and the abilities of a trained decoder on CLIP embeddings. These experiments were conducted for the *Mini-Project 1: Multimodal Learning* of the Advanced Topics in Deep Learning course at Brown University. The datasets used were CIFAR and Flicker 8K [Hodosh et al., Krizhevsky et al.].

1 Introduction

Foundation models have recently gained a significant amount of traction, acting as generalized basic building blocks for whole suites of applications. The purpose of this research is to test the zero-shotting, linear-probing, stitching, and fine-tuning capabilities of common Foundation models. In particular, we experiment with the Contrastive Language-Image Pre-training (CLIP), which was trained on image-caption pairs collected from the Internet. CLIP employs contrastive learning to create a latent space that possesses rich representations for both image and text data [Radford et al., 2021]. In particular, the image encoder we used was a ResNet-50 and the text encoder was a masked self-attention Transformer. Additionally, we use the GPT-3 Davinci, a large decoder-only autoregressive text generator, to generate captions from a list of objects. We experiment with compositions of these two foundation models using natural language, in a procedure similar to the construction of *Socratic Models* [Li et al., 2023]. We also experiment briefly with the effects of basic prompt-engineering on the stitching of these models. The experiments and some significant results are delineated in the sections below.

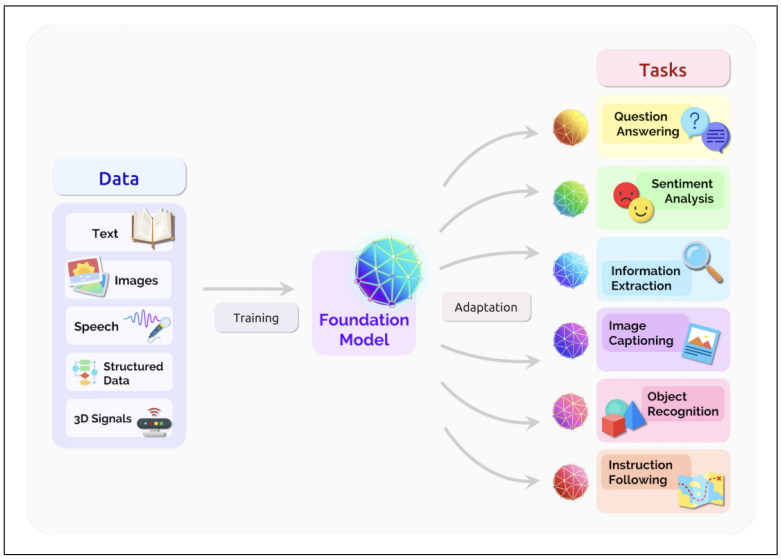

Figure 1: Foundation Models Structure [Bommasani et al., 2022]

2 Zero-Shot Classification

The first experiment I tested was zero-shot classification with CLIP. I randomly sampled 500 examples from the CIFAR 100 Test Set. For each class label in CIFAR 100, I created a sentence in the format *This is a photo of a {LABEL}*. Each class sentence was (1) tokenized and (2) encoded with CLIP's text transformer. Each of the 500 sampled images was encoded with CLIP's Resnet-50. Then, the pairwise cosine similarity was calculated to determine the most similar image and label sentence in CLIP's latent space.

Algorithm 1 Zero-Shot CIFAR-100 Classification with CLIP

Require: Image Tensor x
Require: $C = \{l : l \in \text{CIFAR-100 label space}\}$
  $C' \leftarrow \{l \mapsto$ "This is a photo of a l" $: l \in C\}$
  $x \leftarrow \texttt{preprocess}(x)$
  $x \leftarrow \texttt{encode\_image}(x)$
  $Y \leftarrow 0$
  $s \leftarrow 0$
  for c' in C' **do**
    $c' \leftarrow \texttt{tokenize}(c')$
    $c' \leftarrow \texttt{encode\_text}(c')$
    if $\frac{c' \cdot x}{\|c'\|\|x\|} > s$ **then**
      $Y \leftarrow c'$
    end if
  end for
  $Y \leftarrow$ label of c'

Described above in Algorithm 1 is a simplified version of the algorithm employed to perform zero-shot classification. The actual algorithm was vectorized to improve parallel performance. This procedure resulted in a classification accuracy of 0.406, and a top-5 accuracy of 0.728. Qualitatively, the predicted labels looked similar or within the same semantic category as the label for a significant proportion of the examples. For instance, an incorrectly classified *motorcycle* was predicted to be a *lawnmower*. A large factor in misclassification potentially may have been the resolution of the images. CIFAR-100 images are of size $32 \times 32 \times 3$ while CLIP expects images of size $224 \times 224 \times 3$.

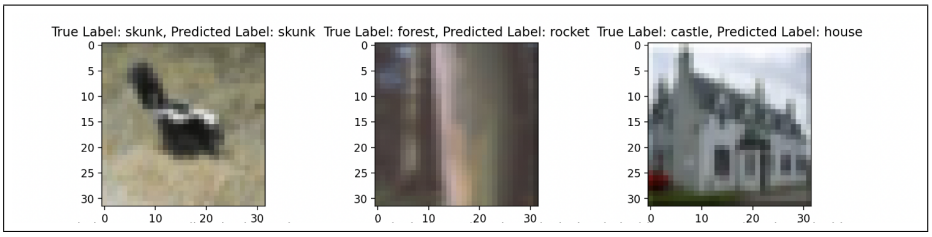

Figure 2: Zero-Shot Classification Examples with CLIP

3 Linear Probing

A linear probing approach to classification was considered next. In this approach, I built a classification head on top of the CLIP image embeddings to classify image label. I used 1000 images to train this classification head. The Adam optimizer and l2 regularization were employed to facilitate training with a learning rate of 0.005 for 50 epochs. Linear probing performs a little better than zero-shot classification, resulting in an accuracy of 0.458 and a top-5 accuracy of 0.772. Qualitatively, results seemed of similar caliber quality to the zero-shot classification model.

There were likely a couple major sources of error: (1) the resolution mismatch between CIFAR-100 images and CLIP inputs likely caused inferior quality embeddings and (2) the small training set size caused significant overfitting due to poor representation from many of 100 classes. The resolution mismatch issue could likely be remedied by using a more intelligent upsampling method, such as a diffusion model. Additionally, a larger training set would likely increase the accuracy of the linear probe.

4 Socratic Method for Image Captioning

While our prior experiments have shown that CLIP has classification potential, we further explore whether stitching CLIP with powerful language generation models can be employed to generated captions for images. In the next experiments, we use the Flicker 8K dataset. I stitched CLIP with GPT-3 to generate captions for the images. The outline of the algorithm I developed is shown in **Algorithm 2.** This algorithm is a combination of linear probing and zero-shot classification, and can be broken into two main steps: (1) converting an image to a list of objects and (2) converting a list of objects to a caption.

Algorithm 2 Socratic Method for Image Captioning

Require: Image Tensor x
Require: Main Object `probe`
Require: Flicker 8K train dataset captions
Require: GPT Prompting Examples P
  $corpus \leftarrow$ `append`(all Flicker 8K captions)
  $common \leftarrow$ `most_common_1500`(all Flicker 8K captions)
  $N \leftarrow \{\}$
  $V \leftarrow \{\}$
  for $w \in common$ **do**
    if `part_of_speech`$(w) \in \{\text{Noun, Proper Noun}\}$ **then**
      $sentence \leftarrow$ "This photo has a " $+ w$
      $N \leftarrow N \cup \{sentence\}$
    else if `part_of_speech`$(w) \in \{\text{Verb}\}$ **then**
      $sentence \leftarrow$ "The action in this photo is " $+ w$
      $V \leftarrow V \cup \{sentence\}$
    end if
  end for

  $N \leftarrow$ `tokenize`(N)
  $N \leftarrow$ `encode_text`(N)
  $V \leftarrow$ `tokenize`(V)
  $V \leftarrow$ `encode_text`(V)

  $x \leftarrow$ `preprocess`(x)
  $x \leftarrow$ `encode_image`(x)
  $main\_obj \leftarrow$ `probe`(x)
  $noun\_cands \leftarrow$ `top_k_cos_sim`(x, N)
  $verb\_cands \leftarrow$ `top_k_cos_sim`(x, V)

  $all\_cands \leftarrow \{main\_obj\} \cup \{noun\_cands\} \cup \{verb\_cands\}$
  $all\_cands \leftarrow$ `remove_similar_candidates`(all_cands)
  $caption \leftarrow$ `prompt_GPT`(P, all_cands)

4.1 Converting an Image to a List of Objects

Toward this goal, I first create a corpus by appending all of the captions together in the Flicker-8K training set. Candidate words are extracted by finding the 1500 most common words and filtering for nouns, proper nouns, and verbs. I convert each of these words to sentences: for nouns, the sentence becomes *"This photo has a {w}"* and for verbs, the sentence becomes *"The action in this photo is {w}"*. This gives me a candidate list of nouns and verbs.

For any image, I use zero-shot classification to find the top 2 most similar nouns and 1 most similar verb via the cosine similarity of the encoding. I remove redundant candidate words by checking whether stemmed candidate words are the same. If so, I remove the word that has a lower cosine similarity with the image.

From exploring the example captions of Flicker-8K, it is evident that there are some very commonly-occurring image objects. Additionally, from manually exploring example captions, our zero-shot classification with CLIP seems biased to capture the more unusual objects in images. Thus, I built a linear probe that classifies whether a dog, boy, man, girl, woman, or child is in the image using 5000 randomly sampled images in the training dataset. I trained this linear probe for 10 epochs and it achieved a test accuracy of 0.732.

An improvement to the current design could involve including adjective candidate descriptors. Furthermore, the distribution of Flicker-8K word counts is long-tailed, which causes issues with the specificity of candidates in the delineated approach. Moreover, the redundancy remover is based off the assumption that redundant words will have the same stem and therefore does not consider synonyms or tenses.

4.2 Converting a List of Objects to a Caption

To convert a list of object to a caption, I use the GPT-3 Davinci text generation model. I use a few-shotting approach to take advantage of in-context learning with three examples of model behavior. These examples are

1. Example('man, cellphone', 'A man is talking on a cellphone outside.')
2. Example('black, dog, running', 'A black dog is running down a field.')
3. Example('men, pond, fishing', 'The men are fishing in a murky pond')

Qualitatively, I found that these examples helped produce the most intelligible results. With other in-context examples, the captioning BLEU score decreased.

4.3 Results

The generated captions mainly captured the meaning of the images. However, the predicted answers would sometimes output null sentences, multiple sentences, or repeated sentences. This behavior is likely a symptom of GPT-3, potentially due to the generation sampling method. Both improved engineering of the prompt as well as improved language generation models could curb this behavior. In addition, the sentences often lacked specific details, only giving a blurry description of the image. This is likely due to the candidate word generation method. We could get better details by segmenting the image into specific sub-images and running the pipeline on the sub-images, but I do believe that there are fundamental limitations to this stitching approach. Even with segmenting, it would be difficult for GPT-3 to piece together the dynamics that are happening in the larger image from sub-images. This limitation boils down to us losing much of the richness of the image when we compress it to a list of descriptor words. I believe that while natural language is a good "glue" for stitching models, a list of objects is fundamentally constrained in description creation.

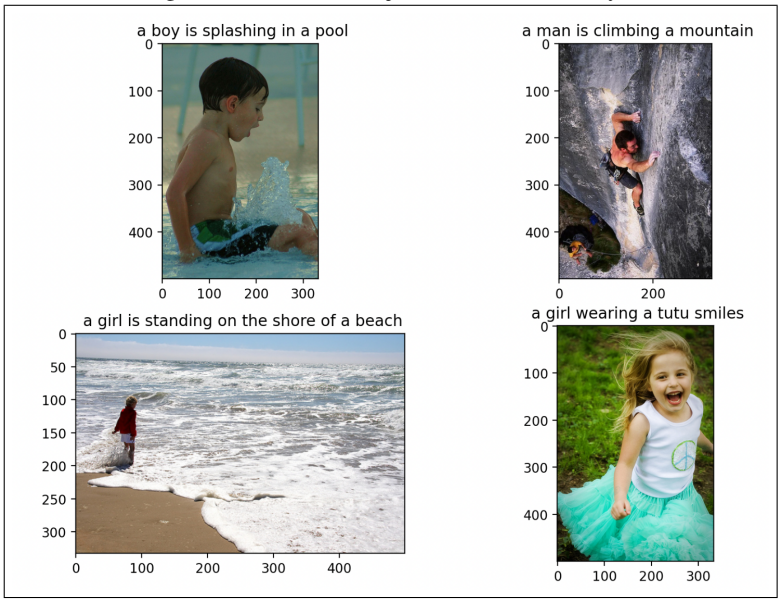

Figure 3: Socratic Generated Captions with high BLEU (> 0.90)

I used the BLEU score to quantitatively score results on a test set of 500. I achieved a BLEU score of 0.433 with an n-gram of 1 and a BLEU score of 0.237 with an n-gram of 2. This performance is significantly worse that the 2019 state-of-the-art on the Flicker 8K data, which achieves a B@1 of 0.699 and B@2 of 0.506 [He and Lu, 2019]. This is likely due to the information bottleneck from using the list of objects. I experimented with tweaking the list of objects for CLIP (by expanding or diminishing the number of most common words to include) and found negligible performance difference. In addition, I experimented with altering the in-context examples are did not find significant

improvements under any circumstances. However, I found that removing the in-context examples decreased the BLEU score significantly (to 0.112 for B@1) and deteriorated caption intelligibility.

5 Captioning Model Improvements with DeCap

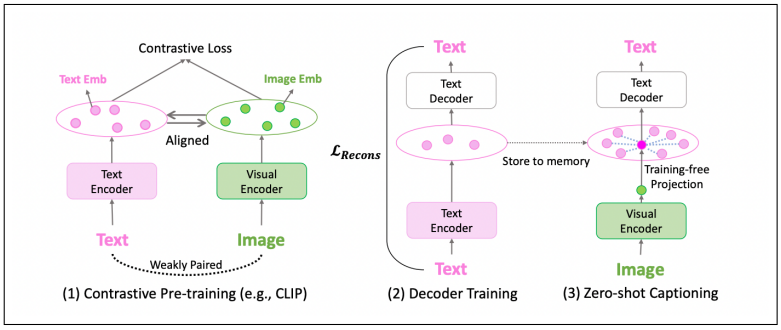

Figure 4: DeCap architecture [Li et al., 2023]

In order to improve the model, I trained a decoder-only transformer on CLIP embeddings to directly caption the images. My architecture was heavily inspired from Li et. al's DeCap (2023). This consisted of projecting image embeddings to CLIP text embedding space using a training-free mechanism and then running the projected embeddings through a decoder-only transformer to generate captions. The decoder-only transformer was trained with a reconstruction loss on the Flicker8K Training Data Set. The projection-based mechanism is calculated as

$$v = \sum_{i=1}^{N} w_i * m_i = \sum_{i=1}^{N} \frac{\exp((m_i^\top v)/\tau)}{\sum_{k=1}^{N} \exp((m_k^\top v)/\tau}$$

where w_i is the weight of ith text embedding in the suport memory [Li et al., 2023]. Essentially, this method reduces the modality gap by taking a weighted average of support text embedding and then running it through the decoder-only transformer.

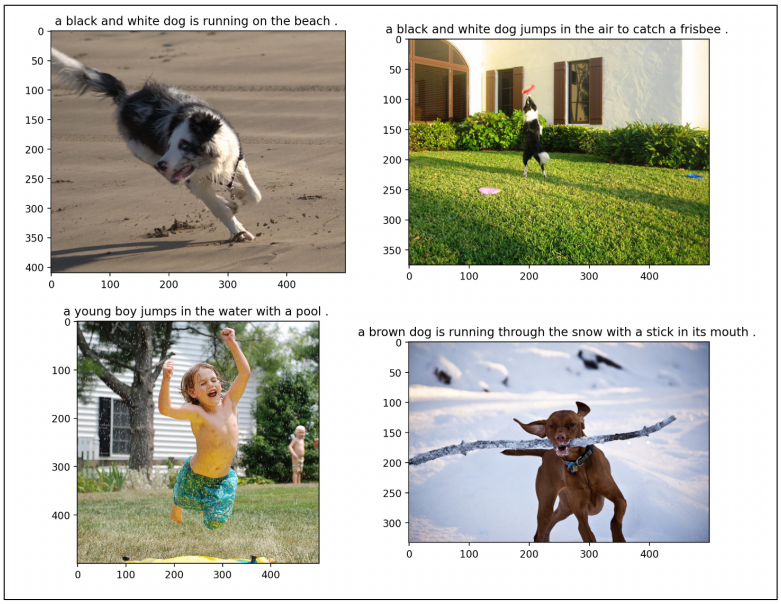

Figure 5: DeCap Generated Captions with high BLEU (> 0.90)

5.1 Decoding CLIP Latents for Zero-Shot Captioning (DeCap)

I trained the decoder-only architecture on the 30000 captions in the Flicker-8K training dataset. This training was very quick as this model only requires CLIP text embeddings to train. Additionally, I used the training dataset as the support memory for the projection method. This method performed much better than GPT-3 stitching, achieving a BLEU score

of 0.568 with n-grams of size 1 and a BLEU score of 0.403 with n-grams of size 2 without any hyperparameter tuning. These measurements were much closer to the state of the art.

6 Discussion

I learned much about the capabilities of foundation models from conducting this research project. CLIP has amazing encoding properties, condensing an expansive state space of all images and all text to a 1024 dimensional space. These representations have very nice geometric properties due to the contrastive learning training procedure. This enables zero shot learning by utilizing distance-based metrics like cosine-similarity. Moreover, representations are rich, leading to useful probing. However, there does exist a modality gap between text and image space, although there are simple methods to project from one space to another. The Socratic Method does show promise, however, it is crucial to choose the right format to represent intermediate state to preserve as much structure through the information bottleneck as possible. A similarly sized model that completes the task end-to-end seems almost always better suited; however, in scenarios when training such a model is too expensive or infeasible, the socratic method seems like a very reasonable option.

7 Acknowledgements

This research project was conducted under instruction from Professor Chen Sun for Mini Project 1. I discussed ideas with Vignesh Pandiarajan.

References

Micah Hodosh, Peter Young, and Julia Hockenmaier. Flickr8k dataset.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-100 (canadian institute for advanced research). URL `http://www.cs.toronto.edu/~kriz/cifar.html`.

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. Learning transferable visual models from natural language supervision, 2021.

Wei Li, Linchao Zhu, Longyin Wen, and Yi Yang. Decap: Decoding CLIP latents for zero-shot captioning via text-only training. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=Lt8bMlhiwx2`.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models, 2022.

Shan He and Yuanyao Lu. A modularized architecture of multi-branch convolutional neural network for image captioning. *Electronics*, 8(12), 2019. ISSN 2079-9292. doi:10.3390/electronics8121417. URL `https://www.mdpi.com/2079-9292/8/12/1417`.