Reimplementing ClipCap: CLIP Prefix for Image Captioning

Peiyan Song, Veronika Grytsai, Patrick Rui De Jing
Brown University

Introduction

We aim to explore how powerful pre-trained vision and language models like **CLIP** and **GPT-2** can be used for image captioning without fine-tuning the large models themselves. This project is a reimplementation of "**ClipCap: CLIP Prefix for Image Captioning**" by Ron et al.s We chose this paper because it proposes an efficient and elegant solution to bridge vision and language using minimal training—just a lightweight Transformer that maps visual features from **CLIP** into **GPT-2's** language embedding space..

Methodology|Architecture|Design

What is CLIP?

CLIP (*Contrastive Language-Image Pre-training*) is a multimodal model developed by OpenAI that can understand and relate images and textual descriptions in a unified manner. Here in our model, CLIP takes an image as input and outputs a single embedding vector.

Dataset

We are using a subset (56674 captions 33,841 for train 12,505 for val) of the **COCO Captions Dataset**, which contains over 330K images, five independent human generated captions are be provided for each image.
Flickr30k dataset is comprised of 31,783 images that capture people engaged in everyday activities and events. Each image has a descriptive caption.

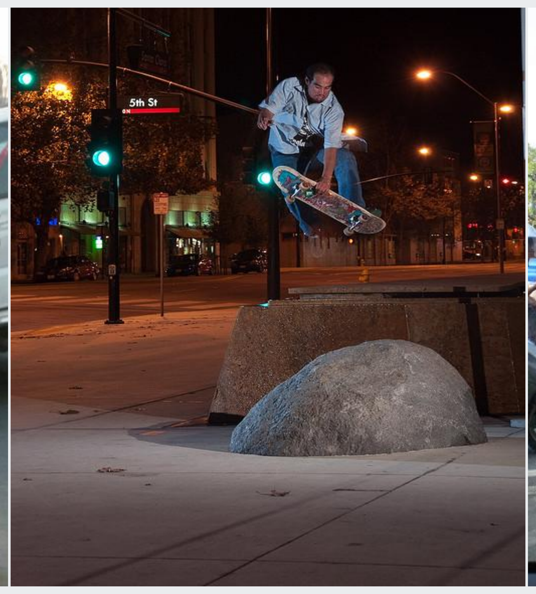
Structure Overview

- CLIP to extract image embeddings.
- A small Transformer (or MLP) (prefix mapper) to convert those embeddings to a sequence of prefix tokens.
- GPT-2 that receives these prefix tokens and continues to generate a caption.

Loss

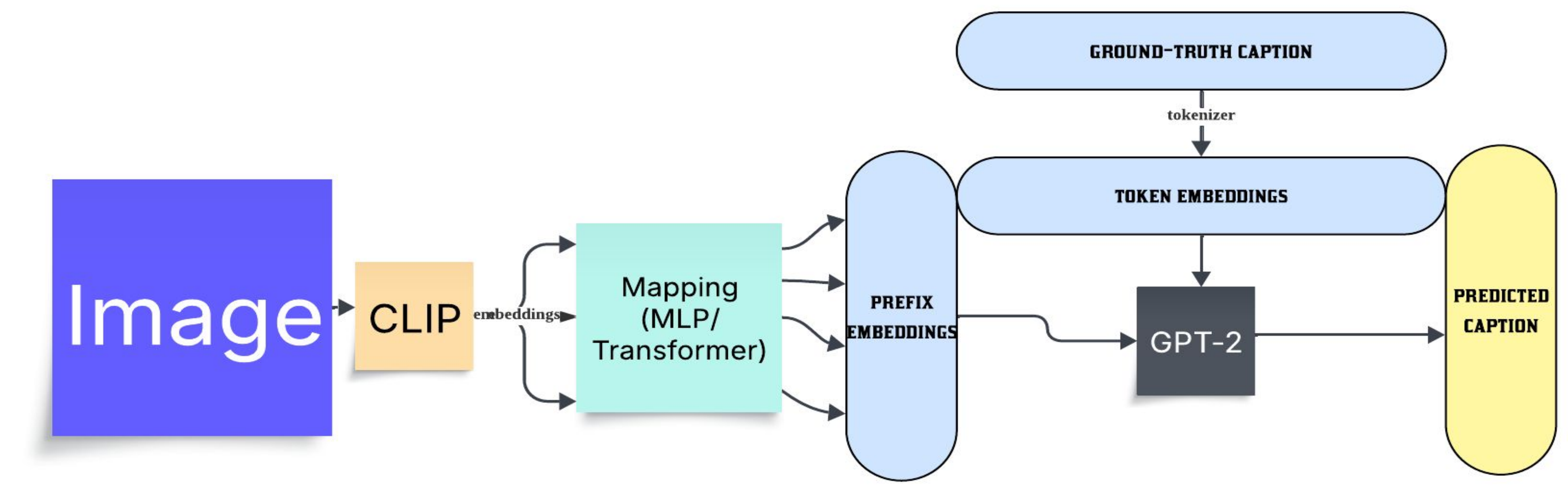
The loss function used here is cross-entropy loss, computed between the predicted token logits and the ground-truth token sequence.

Results



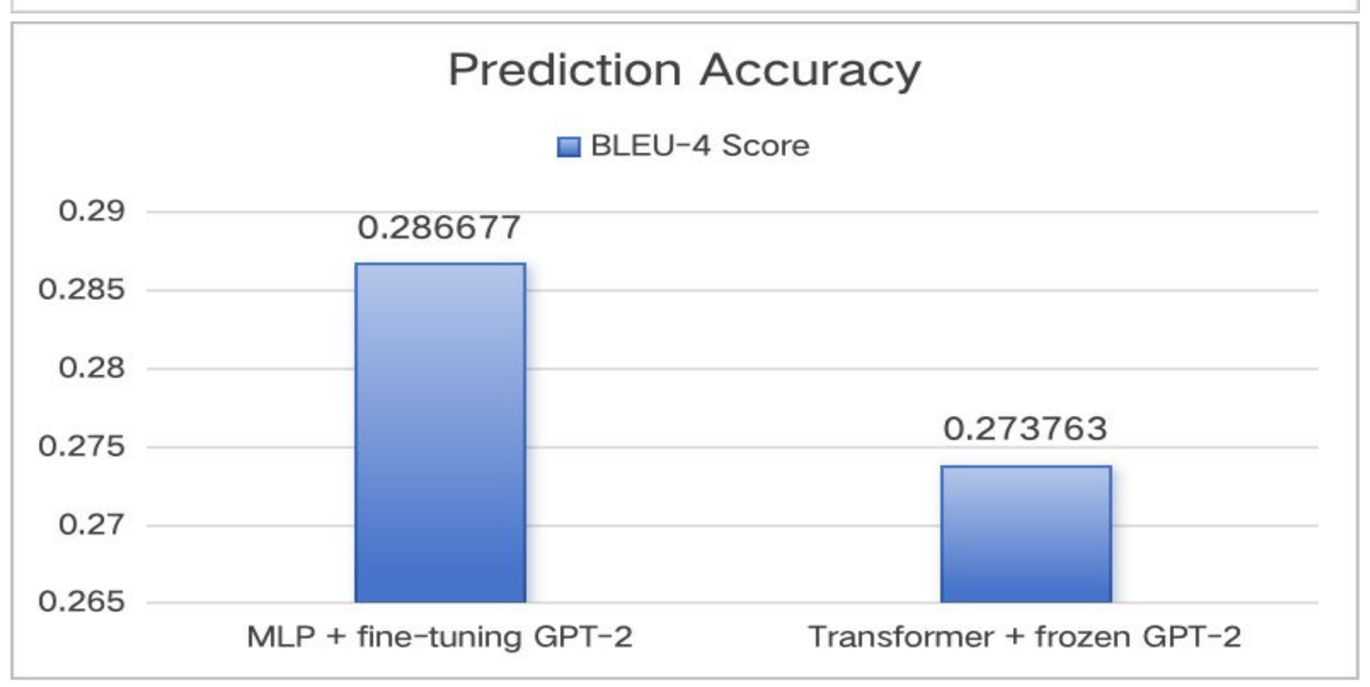
Ground Truth	A yellow white and red bus lost a wheel.	A skate boarder doing jumps at night on city street.	Many different people parked on motor bikes on a street.
MLP + fine-tuning GPT-2	A white and red bus driving down a street.	A man on a skateboard doing tricks on a ramp.	A group of people riding motorcycles down a street.
Transformer + frozen GPT-2	A white and red truck is parked on a street.	A skateboarder doing a trick on a skateboard.	A group of people on motorcycles on a street.

Architecture Diagram



Evaluation

- This figure shows how the loss changes with epochs during training.
- We observe that using a Transformer mapper with a frozen GPT-2 leads to faster convergence — the model reaches a low training loss more quickly. This is likely because the frozen GPT-2 provides stable and consistent language generation, allowing the mapper to adapt rapidly without destabilizing the overall model.
- On the other hand, the MLP mapper combined with a fine-tuned GPT-2 achieves a lower eventual loss. Although it converges more slowly, fine-tuning GPT-2 allows the model to better adapt to our specific dataset, leading to improved performance in the long run.



- BLEU-4 is a metric for evaluating machine-generated text by comparing it to reference translations using 1-gram to 4-gram precision. It penalizes short translations and outputs a score between 0 and 1, with higher scores indicating better quality.
- The evaluation compares BLEU-4 scores for captions generated from 193 images using two models. The MLP model with fine-tuned GPT-2 achieved a higher average BLEU-4 score of 0.287, indicating better alignment with reference captions, while the Transformer model with frozen GPT-2 scored 0.274, suggesting slightly lower performance.

Challenges

- OpenAI's original CLIP model (ViT, RN50, RN101 backbones) was released in PyTorch. Therefore, switching it to Tensorflow including using HuggingFace's TFCLIP (Clip-like package in TF).
- Training the model using a subset of COCO dataset took ~10 hours
- Choosing evaluation metrics
- Implementing MLP vs Transformers

Limitations

- **No fine-tuning of CLIP and GPT-2** – The vision and language models are frozen; only a small **Transformer (or MLP)** is trained.
- **Only supports image → text** – No bidirectional tasks like text → image or visual question answering.
- **Basic captions only** – Can't handle multi-sentence descriptions, storytelling, or dialog.
- **Relies heavily on CLIP** – If CLIP misses visual info, the model can't recover it.