Image Caption Generation With CLIP+GPT-2 Model

Siyuan Jing and Haonan Wu
Boston University
siyuan16@bu.edu and whn17@bu.edu

May 3, 2025

Abstract

To be finished

1 Introduction

Image caption generation, the task of recognizing images to generate natural language descriptions, lies at the critical intersection of computer vision and natural language processing. It plays an important role in a variety of applications, including image retrieval, accessibility for the visually impaired, and automated image content processing.

Recent advances in vision-language models have enabled more accurate and fluent caption generation by leveraging large-scale pretraining on aligned image-text pairs . These models demonstrate remarkable capabilities in understanding visual content and translating it into coherent text descriptions.

Motivated by this progress, we focus our project on replicating a CLIP+GPT-2-based image captioning system, with inspriation from Nukrai et al.[1]. Through this project, we seek to explore and deepen our understanding of machine learning, as well as its broader applications in fields such as computer vision and natural language generation.

Our goal is to:

- Leverage the pretrained CLIP model to extract semantically rich image embeddings without the need for training a custom vision encoder.
- Utilize the generative capabilities of GPT-2 to produce fluent and coherent natural language captions.
- Bridge the gap between visual and textual modalities by introducing a projection layer that maps image embeddings into GPT-2's input space.
- Enable flexible and data-efficient image captioning, where the visual semantics guide the generation through prefix-based conditioning.
- Evaluate the quality of the generated captions using standard metrics such as BLEU and CIDEr, in order to quantitatively assess the model's accuracy and relevance.

2 Background and Related Work

2.1 Contrastive Language-Image Pretraining(CLIP)

CLIP from OpenAI is a visual-language model. Instead of relying on task-specific supervised learning, CLIP is trained on a dataset of 400 million image-text pairs collected from the internet using a contrastive loss function. CLIP consists of two separate encoders: a visual encoder (ResNet) for images, and a text encoder (Transformer) for captions. Its ability to generate rich, semantically meaningful image embeddings makes CLIP a powerful foundation for our systems and an ideal visual component in our CLIP+GPT-2 image captioning pipeline. For example, according to Mokady et al.[2], it mentioned that the visual encoding capability of CLIP can be used to embed and project the generated images into the input space of GPT-2 to generate prefixes, which helps the final caption generation of GPT-2. Inspired by this article, we decided to study the CLIP architecture and implement related deployments.

2.2 Generative Pre-trained Transformer 2 (GPT-2)

GPT-2 is a large-scale language model based on the Transformer decoder architecture proposed by Radford et al.[3]. According to the paper, Transformer completely replaces the traditional RNN or CNN structure with self-attention, which is more efficient and accurate when processing long sequence dependencies. Therefore, we consider implementing the transformer structure as our decoder of the whole pipeline. Unlike the traditional Transformer, which contains both encoder and decoder components, GPT-2 uses only a decoder. This design enables the model to predict the next token based solely on previously generated tokens, making the generated text semantically relevant and well suited for text generation tasks such as image captioning.

2.3 Multi-Head Attention Mechanism

Inspired by the *"Attention Is All You Need"* paper [4], we employ a multi-head attention mechanism as a cross-modal connector between the CLIP image embedding and the GPT-2 language model. Instead of using a linear layer to directly project the CLIP features into the GPT-2 embedding space, we employ an attention mechanism that enables the model to selectively focus on different aspects of the visual features when generating each token in the caption, making the generated tokens more consistent with the image information. Multi-head Attention network allows the model to learn how different regions in the image affect language production.

We compare these advanced approaches of image caption generation to understand its advantages and limitations, and implement these methods in the process of model training to get our own model.

3 Methodology

3.1 Implementation Details

Detail of Implementation and pipeline: This is the pipeline of our entire training process, which mainly includes multiple steps: image input, feature extraction and encoding, feature processing, feature decoding, model training and error evaluation. We will also introduce the optimization we used.

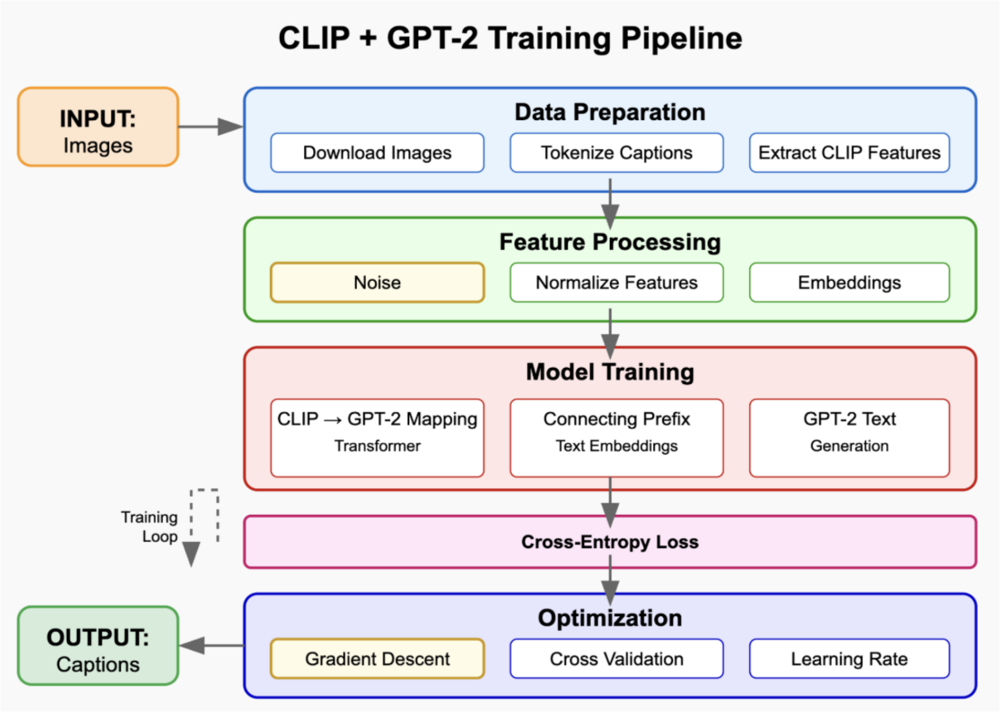

Figure 1: CLIP+GPT-2 Training Pipeline

3.1.1 Data Set

We utilize the MS COCO dataset [5]for training and evaluation, MS COCO (Microsoft Common Objects in Context) is a large-scale image recognition, segmentation, and annotation dataset that is widely used in research in the field of computer vision, especially in tasks such as image captioning, object detection, and semantic segmentation. It contains 330K images with 5 captions each. We will split our dataset into three parts: 80% training, 10% validation, 10% testing.

3.1.2 Data Preparation

After we get the original CLIP embedding, we will normalize it for better training performance. In the actual training process, we found that the trained model may be overly sensitive to some features of some pictures, so we add some noise to 10% of the feature information of each picture to improve generalization ability and reduce overfitting. These embeddings can then be projected into the GPT-2 space for training.

3.1.3 Model training

Next, we use a multi-layer Transformer to map the CLIP embedding to the word embedding space of GPT-2 to create a prefix. This prefix is then concatenated with the tokenized caption, which is generated in the data preparation stage, to form the whole input sequence for GPT-2. After receiving these input sequences, the GPT-2 model is trained for a set rounds of training loop to generate the caption of the corresponding image based on the given prefix and token.

3.1.4 Optimization method

For the weight matrix obtained after each round of training, we use sub-gradient descent to adjust the weights. Additionally, during the entire training process, we use cross-validation to find the optimal solution for multiple hyperparameters in the model, and reasonably control the learning rate to obtain better training results.

3.2 Loss and objective function for training process

We use the cross-entropy loss function to measure the difference between the caption generated by the model and the ground-truth captions. The loss function is defined as:

The Cross-entropy Loss Function:

$$\text{Loss} = -\sum_{t=1}^{T} \log P_{\text{GPT-2}}(w_t \mid w_{<t}, \text{prefix}) \tag{1}$$

Where:

- T — the total number of tokens (words) in the caption
- w_t — the actual word at position t in the ground-truth caption
- $w_{<t}$ — the sequence of all previous words before time step t
- prefix — the mapped image feature vector used to condition the generation
- $P_{\text{GPT-2}}(w_t \mid w_{<t}, \text{prefix})$ — the probability assigned by GPT-2 to word w_t given previous words and the image prefix

This loss encourages the model to assign a higher probability to the correct next word at each time step. Minimizing the cross-entropy loss over the training data helps the model generate captions that are more fluent and accurate.

The Objective Function in Maximum Likelihood Estimation:

$$\hat{\theta} = \arg\min_{\theta} - \sum_{i=1}^{N} \sum_{t=1}^{T_i} \log P_\theta(w_t^{(i)} \mid w_{<t}^{(i)}, \text{prefix}^{(i)}) \tag{2}$$

Where:

- θ — model parameters, including GPT-2 weights and the parameters of the prefix mapping network
- N — the total number of training samples (image-caption pairs)
- T_i — the number of tokens in the i-th caption
- $w_t^{(i)}$ — the t-th word in the i-th ground-truth caption
- $w_{<t}^{(i)}$ — all previous words before position t in the i-th caption
- $\text{prefix}^{(i)}$ — the mapped image features for the i-th image

This objective function aims to find the parameter set $\hat{\theta}$ that minimizes the total negative log-likelihood across the training dataset. During training, the model updates θ using optimization techniques such as gradient descent, guided by cross-validation to avoid overfitting and improve generalization.

Training Details: We train our model using these hyperparameter after cross-validation:

- Prefix length: 40
- Transformer layers: 8
- Optimizer: Adam with a learning rate of 5×10^{-5}
- Number of epochs: 20
- Noise rate: 10%

3.3 Evaluation Protocol

We evaluate the quality of learned representations by training a simple linear classifier on top of the frozen embeddings.

4 Experiments and Results

4.1 Dataset and Preprocessing

We conduct experiments on:

- **CIFAR-10:** A 10-class dataset with 60,000 images.
- **STL-10:** A larger dataset often used for unsupervised learning benchmarks.

4.2 Baseline Comparisons

We compare SimCLR embeddings with:

- PCA-based dimensionality reduction ($d = 128$)
- Autoencoders trained on the same dataset
- Supervised ResNet-18 trained on CIFAR-10

Method	CIFAR-10 Accuracy (%)	STL-10 Accuracy (%)
Supervised ResNet-18	92.5	85.4
PCA + kNN	45.6	38.2
Autoencoder + kNN	55.3	49.6
SimCLR (Ours)	80.2	76.4

Table 1: Comparison of representation learning methods. SimCLR significantly outperforms classical techniques.

4.3 Ablation Studies

Effect of Temperature τ: We analyze how different values of τ in the contrastive loss impact performance.

5 Discussion

5.1 Key Findings

- SimCLR significantly outperforms PCA and autoencoders in feature learning.
- The choice of augmentations greatly affects performance.
- Higher temperature values in contrastive loss lead to better separation of features.

5.2 Future Work

- Extend to other self-supervised methods (e.g., BYOL, MoCo).
- Apply to domain adaptation tasks.
- Explore contrastive learning for text or multimodal applications.

6 Conclusion

Our empirical study demonstrates the effectiveness of contrastive learning via SimCLR for representation learning. By systematically evaluating augmentation pipelines, batch sizes, and loss functions, we provide insights into optimizing contrastive learning for different datasets.

References

[1] Nukrai, D., Mokady, R., & Globerson, A. (2022). Text-Only Training for Image Captioning using Noise-Injected CLIP. https://doi.org/10.48448/n7sq-p557

[2] Mokady, R., Hertz, A., & Bermano, A. H. (2021). ClipCap: CLIP Prefix for Image Captioning. http://arxiv.org/abs/2111.09734

[3] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). "Language Models are Unsupervised Multitask Learners." OpenAI Technical Report.

[4] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). "Attention is all you need." Advances in Neural Information Processing Systems, 30.

[5] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. 2015. Microsoft coco captions: Data collection and evaluation server. arXiv preprint arXiv:1504.00325.

[6] Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002). "BLEU: a method for automatic evaluation of machine translation." ACL 2002.

[7] Vedantam, R., Lawrence Zitnick, C., & Parikh, D. (2015). "CIDEr: Consensus-based image description evaluation." CVPR 2015.

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, Geoffrey Hinton. "A Simple Framework for Contrastive Learning of Visual Representations." ICML 2020.

[9] Geoffrey Hinton, Ruslan Salakhutdinov. "Reducing the Dimensionality of Data with Neural Networks." Science, 2006.