Domain Adaptation of CLIP for Multimodal Visual Tasks: Captioning and Visual Question Answering

Mayukh Sen (ms3802) Rohit Kulkarni (rk1169)

Abstract

This project explores the adaptation of the CLIP model for two vision-language tasks: Visual Question Answering (VQA) and Image Captioning. For VQA, the *Visual Genome* dataset was used, while *Laion* and *Flickr30k* were utilized for Captioning. Domain adaptation with adversarial methods bridged the gap between pretraining and task-specific datasets. Empirical evaluations using *BLEU* and *accuracy* metrics highlight improved model performance.

1 Problem Description

1.1 Visual Question Answering (VQA)

The task of Visual Question Answering (VQA) involves generating accurate responses to textual questions based on the visual content of an image. This requires learning a mapping from the joint space of image-text features to a set of answers.

Given an image I and a question Q, the goal is to predict the most probable answer A:

$$\hat{A} = \arg\max_{A} P(A \mid I, Q)$$

where $P(A \mid I, Q)$ is modeled as a combination of the visual features extracted from I and the textual features derived from Q.

- **Image Encoding:** The image $I \in R^{H \times W \times C}$ is encoded into a feature vector $\mathbf{v}_I \in R^d$ using an image encoder:
- **Text Encoding:** The question $Q = \{q_1, q_2, \ldots, q_m\}$ is encoded into a feature vector $\mathbf{v}_Q \in R^d$ using a text encoder:
- **Feature Fusion:** The joint representation $\mathbf{v}_{I,Q} \in R^d$:

$$\mathbf{v}_{I,Q} = f_{\text{fusion}}(\mathbf{v}_I, \mathbf{v}_Q)$$

- **Answer Prediction:** A classifier predicts the probability distribution over possible answers:

$$P(A \mid I, Q) = \text{softmax}(\mathbf{W} \cdot \mathbf{v}_{I,Q} + \mathbf{b})$$

The predicted answer is:

$$\hat{A} = \arg\max_{A} P(A \mid I, Q)$$

1.2 Image Captioning

Image Captioning involves generating descriptive textual captions for given images. This is formulated as a sequence prediction task.

Given an image I, the task is to generate a caption $C = \{w_1, w_2, \ldots, w_n\}$, where w_i represents the words in the caption. The objective is to maximize the probability of the caption given the image:

$$P(C \mid I) = \prod_{i=1}^{n} P(w_i \mid w_1, w_2, \ldots, w_{i-1}, I)$$

- **Image Encoding:** The image I is encoded into a feature vector $\mathbf{v}_I \in R^d$ or a sequence of feature vectors $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$
- **Sequence Generation:** At each time step t, the model predicts the next word w_t:

$$P(w_t \mid w_1, w_2, \ldots, w_{t-1}, I) = \text{softmax}(\mathbf{W} \cdot \mathbf{h}_t + \mathbf{b})$$

where $\mathbf{h}_t \in R^d$ is the hidden state of the sequence decoder at time t:

$$\mathbf{h}_t = f_{\text{decoder}}(w_{t-1}, \mathbf{h}_{t-1}, \mathbf{v}_I)$$

- **Output Sequence:** The sequence C is generated iteratively:

$$\hat{w}_t = \arg\max_{w_t} P(w_t \mid w_1, w_2, \ldots, w_{t-1}, I)$$

1.3 Domain Adaptation in CLIP

Contrastive Loss: CLIP is trained using a contrastive loss to align the embeddings of matched (image, text) pairs:

$$\mathcal{L}_{\text{contrastive}} = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{\exp(\mathbf{v}_{I_i} \cdot \mathbf{v}_{Q_i}/\tau)}{\sum_{j=1}^{N} \exp(\mathbf{v}_{I_i} \cdot \mathbf{v}_{Q_j}/\tau)}$$

where τ is a temperature parameter, N is the batch size, and $\cdot$ denotes the cosine similarity between embeddings.

Adversarial Domain Adaptation: For domain adaptation, an adversarial loss aligns feature distributions between pretraining and task-specific datasets:

$$\mathcal{L}_{\text{ADA}} = E_{x \sim D_{\text{source}}}[f(x)] - E_{x \sim D_{\text{target}}}[f(x)]$$

where $f(x)$ is a discriminator that distinguishes between source and target domains.

2 Model Description

2.1 Contrastive Language–Image Pretraining (CLIP)

CLIP is a multimodal model that aligns images and text into a shared embedding space. Its architecture comprises:

- **Image Encoder**: Processes images using models like Vision Transformers (ViT) or ResNet to produce image embeddings ($\mathbf{v}_I$).
- **Text Encoder**: A Transformer-based model maps tokenized text to text embeddings ($\mathbf{v}_T$).

The encoders are jointly trained with contrastive loss on a large dataset of image-text pairs. The architecture supports:

1. **Contrastive Pre-training**: Jointly trains image and text encoders to align embeddings for paired inputs.
2. **Zero-shot Prediction**: Enables direct use of textual prompts as classifiers for image classification tasks without task-specific fine-tuning.

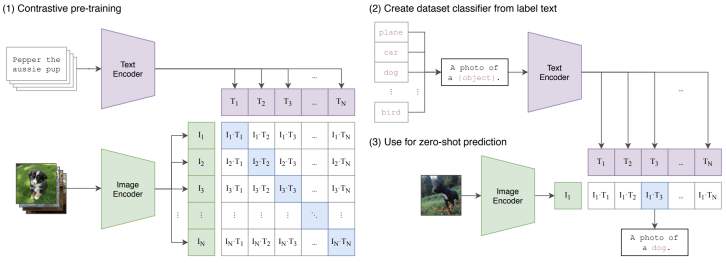

Figure 1. Summary of our approach. While standard image models jointly train an image feature extractor and a linear classifier to predict some label, CLIP jointly trains an image encoder and a text encoder to predict the correct pairings of a batch of (image, text) training examples. At test time the learned text encoder synthesizes a zero-shot linear classifier by embedding the names or descriptions of the target dataset's classes.

Figure 1: CLIP architecture: Joint pretraining of text and image encoders using contrastive loss, enabling multimodal alignment and zero-shot predictions.

2.2 Model for VQA

The dataset for Visual Question Answering (VQA) consists of images, questions, and corresponding answers. Each image is preprocessed, and the questions are tokenized using a CLIP-compatible processor.

The `VQA_CLIP` model extends the CLIP model for VQA with the following components:

- **CLIP Backbone:** Extracts embeddings for input text and images.
- **Classifier:** A fully connected neural network maps image embeddings to the answer space. It dynamically adjusts dimensions to match the number of answer classes and includes dropout for regularization and ReLU activation for non-linearity.

During the forward pass:

- The model takes tokenized questions and image embeddings as inputs.
- It predicts logits for the possible answers.

A vocabulary is created to map unique answers to indices. This facilitates efficient handling of answers during training and evaluation by translating answers into a numerical format. The cosine similarity loss is used to align predicted and target embedding. Normalization ensures predictions and targets are unit vectors. We measure the alignment between these embeddings, penalizing misaligned predictions.

2.2.1 Hyperparameters and Training Optimization

The model is trained using a batch size of 32, a learning rate of 5×10^{-5} with the AdamW optimizer, and a weight decay of 10^{-4} to prevent overfitting. Training spans 3 epochs with gradient accumulation steps set to 2 for handling larger effective batch sizes. A step-based learning rate scheduler is used to improve convergence, and mixed precision training optimizes memory usage and speeds up computations. The model is trained on an A100 GPU (`cuda`), with efficient data loading to maximize hardware utilization.

2.3 Model for Captioning

The model utilizes two datasets for training and fine-tuning. LAION is used for pretraining, providing large-scale image-text pairs as precomputed embeddings and captions. Flickr30k is used for fine-tuning, focusing on high-quality captions aligned with their corresponding images.

The Captioning model extends the CLIP backbone with a Transformer-based decoder for generating captions. The CLIP backbone extracts fixed-size embeddings from images to represent their semantic content. A six-layer Transformer decoder then processes these embeddings to generate textual descriptions, with captions tokenized and embedded before being passed to the decoder. Finally, a projection layer maps the decoder outputs back into the embedding space to predict the next token embeddings.

2.3.1 Adversarial Domain Adaptation

The model incorporates a domain discriminator to classify embeddings as belonging to either the source (LAION) or target (Flickr30k) domains. Through adversarial training, the model learns to confuse the discriminator, effectively aligning embeddings from both domains and improving generalization across datasets.

2.3.2 Training Methodology

The model is pretrained on LAION embeddings using Mean Squared Error (MSE) loss between predicted and actual caption embeddings. Fine-tuning is performed on Flickr30k to achieve better alignment with high-quality captions. Additionally, adversarial training is applied to ensure domain invariance by aligning embeddings from both datasets.

3 Emperical Evaluation Of Models

3.1 Captioning Model

The training dataset is prepared by embedding images and captions from the LAION dataset. The model is pre-trained for one epoch without adversarial domain adaptation (ADA), achieving a **training loss** of **0.0068**. After loading the pre-trained model and continuing for another epoch, the **training loss** further reduces to **0.0022**, reflecting effective learning of image-caption relationships. With ADA, the model undergoes **10 epochs** of adversarial training, during which the discriminator loss decreases from **0.2982** to **0.0918**, and the adversarial loss increases from **0.3024** to**0.9251**, indicating improved domain alignment through adversarial optimization.

The model is subsequently fine-tuned on the Flickr30K dataset with ADA. Over **10** epochs, the discriminator loss continues to decrease, starting from **0.1689** and reaching **0.0918**, while the adversarial loss steadily increases from **0.5290** to **0.9251**. When evaluated using BLEU scores, the model achieves a score of **0.2805** without ADA, while fine-tuning with ADA improves the BLEU score to **0.4116**. This significant improvement highlights the effectiveness of adversarial domain adaptation in enhancing the model's ability to generate high-quality, domain-specific captions, despite some signs of overfitting in the adversarial component.

3.2 VQA Model

The training dataset is created by extracting objects and relationships from their respective JSON files. The model is trained to learn relationships between objects . It is trained on 344749 samples.

After training for 3 epochs the model it achieves training loss (1- cosine similarity) of **0.33**. This indicates the model is able to learn relationships between identified objects quite well . However the model overfits slightly as it achieves a validation loss of 0.6.Despite the overfitting , it performs VQA quite effectively on toy examples. It does however outperform zero shot CLIP performance.

3.2.1 Examples

Figure 2: Correctly identifies a watermelon poster and gives a true relationship despite grammatical mistakes.

Figure 3: Correctly identifies orange.

Figure 4: Recognizes sofa and mattress effectively, demonstrating VQA.

4 Advantages and Disadvantages Of Models

4.1 Captioning Model

4.1.1 Advantages

- Model leverages ADA to significantly improve BLEU scores, achieving a score of 0.4116 compared to 0.2805 without ADA. This highlights the ability to adapt to domain-specific data effectively
- Hyperparameter tuning and the use of discriminator and adversarial losses improve image-caption alignment, shown by the steady reduction in discriminator loss
- Model generates accurate captions, excels in domain-specific contexts, and outperforms baseline metrics.

4.1.2 Disadvantages

- Rising adversarial loss during fine-tuning indicates minor overfitting, potentially limiting generalization
- Multi-epoch adversarial training and fine-tuning are resource-intensive, requiring significant computational power and time, making it less practical for low-resource environments. A possible alternative could be using knowledge distillation or model compression techniques to reduce computational requirements

4.2 VQA Model

4.2.1 Advantages

- Obtains very good training loss and is able to outperform Zero-Shot CLIP in VQA
- Trained effectively using parallel processing , gradient accumulation. Able to bring faster convergence using hyperparameter tuning and batch sizes of good numbers
- Robust enough to discern single and multiple objects and stick to object recognition for single objects

4.2.2 Disadvantages

- Overfitting due to very large training dataset and compute intensive training means significant resource expenditure
- Very large object space means it is hard for model to select the correct item from the huge probability distribution in object space(75000 unique objects). Need for sophisticated searching methods like Beam Search employed in LLMs to optimize and prevent mistakes

References

[1] Alec Radford, Jong Wook Kim, Chris Hallacy, et al. *Learning Transferable Visual Models From Natural Language Supervision.* arXiv preprint arXiv:2103.00020, 2021. https://arxiv.org/abs/2103.00020.

[2] Yaroslav Ganin and Victor Lempitsky. *Unsupervised Domain Adaptation by Backpropagation.* arXiv preprint arXiv:1702.05464, 2016. https://arxiv.org/abs/1702.05464.

[3] Chenyu Wang, Chen Wei, Alan Yuille, et al. *Contrastive Language-Image Pretraining for Domain Generalization.* arXiv preprint arXiv:2203.05557, 2022. https://arxiv.org/abs/2203.05557v2.

[4] Minghao Xu, Jian Zhang, Bingbing Ni, Teng Li, Chengjie Wang, Qi Tian, Wenjun Zhang. *Adversarial Domain Adaptation with Domain Mixup.* Shanghai Jiao Tong University, Huawei Hisilicon, Anhui University, Youtu Lab, Tencent, Huawei Noah's Ark Lab, 2020. https://arxiv.org/pdf/1912.01805

[5] Kaiming He, Haoqi Fan, Yuxin Wu, et al. *Momentum Contrast for Unsupervised Visual Representation Learning.* arXiv preprint arXiv:1912.01805, 2019. https://arxiv.org/abs/1912.01805.

5 Appendix

Google Drive Link
*did not give Github as files were hard to upload there and we used Google Colab to train