CLIP Implementation

Corentin Clément, Victor Buthod, Guillaume Charvolin

19 fevrier 2024

Majeure SCIA-G

EPITA

Table des matières

1 Introduction

The CLIP (*Contrastive Language–Image Pretraining*) model proposed by OpenAI has revolutionized the way shared visual and textual representations are learned. The core idea is to align two latent spaces—one for images and one for text—enabling tasks such as retrieving images based on textual descriptions or automatically describing images.
In this project, we reproduce and adapt the approach described in the original paper. Three architecture variants were explored :

- **BERT + ResNet :** Combining BERT for text and ResNet for images.
- **BERT + ViT :** Using BERT for text and ViT (Vision Transformer) for images.
- **GPT + ViT :** Employing GPT-2 for text and ViT for images.

Each model was trained using a contrastive loss function to maximize similarity between matching image-text pairs and minimize similarity between non-matching pairs.

2 Dataset and Preprocessing

2.1 Dataset Description

The dataset used, **Fashion Product Images Small**, consists of approximately 44,000 product images paired with textual descriptions from the `productDisplayName` column. These descriptions are typically simple, such as "Red Men's T-Shirt." While this dataset is suitable for basic tasks, it has limitations :

- Text descriptions are often repetitive or vague, making it difficult to distinguish between images.
- The lack of contextual or complex information may hinder models like GPT, which require richer inputs.

2.2 Data Preprocessing

The preprocessing steps included :

- Resizing images to 224 × 224 to be compatible with ResNet and ViT architectures.
- Normalizing pixel values to the range [-1, 1].

- Using raw textual descriptions without additional enrichment.

Unlike the original CLIP approach, we did not create separate training and testing splits due to resource and time constraints.

3 Model Architecture

Each CLIP model consists of two independent encoders : one for images and one for text. These encoders project inputs into a shared latent space of dimension 128, followed by L2 normalization. A learnable temperature parameter adjusts the similarity distribution during training.

3.1 Image Encoders

Two architectures were tested for encoding images :

- **ResNet :** A convolutional model pre-trained on ImageNet.
- **ViT :** A Vision Transformer that processes images as sequences of patches.

3.2 Text Encoders

For text encoding, two models were evaluated :

- **BERT :** A bidirectional model pre-trained on language understanding tasks.
- **GPT-2 :** A generative model capable of producing sequences of text.

4 Experiments and Results

4.1 Training

All three models were trained for 5 epochs on the entire dataset with a batch size of 64. Training was conducted on Google Colab, limiting the number of epochs and the ability to explore additional hyperparameters.

4.2 Evaluation

Performance was evaluated using two metrics :

- **Accuracy :** The percentage of image-text pairs where the correct text had the highest similarity.
- **Top-100 Accuracy :** The percentage of pairs where the correct text was among the top 100 most similar texts.

Model	**Accuracy (%)**	**Top-100 Accuracy (%)**
BERT + ResNet	0.74	34.51
BERT + ViT	0.87	37.11
GPT + ViT	0.00	0.10

TABLE 1 – Performance comparison of the three tested models.

4.3 Result Analysis

BERT + ResNet and **BERT + ViT** showed superior performance, with BERT + ViT achieving slightly better results. This can be attributed to :

- **BERT :** Pre-trained on language understanding tasks, it is well-suited for simple textual descriptions.
- **ViT :** Outperforms ResNet by capturing finer visual details through its patch-based approach.

In contrast, **GPT + ViT** failed to achieve meaningful results. Possible explanations include :

- **Simplistic text descriptions :** GPT, designed for complex sequences, struggled with basic inputs.
- **Alignment complexity :** GPT's generative nature complicates its alignment with visual representations.
- **Undertraining :** Five epochs were insufficient for such a complex architecture to converge.

5 Simplifications and Limitations

Several simplifications were made to address resource constraints :

- **Simplistic textual input :** Direct use of `productDisplayName`.
- **No train/test split :** The model was evaluated on the same data used for training.
- **Limited epochs :** The small number of epochs impacted the model's ability to converge.

These simplifications partly explain the modest performance, particularly for GPT + ViT.

6 Conclusion

This project successfully reproduced a simplified version of CLIP and tested three architectures. Results indicate that BERT + ViT is the best combination in this context, balancing efficiency and accuracy. Future improvements could include :

- Enriching the dataset with more descriptive and diverse textual annotations.
- Extending training time and exploring hyperparameter tuning.
- Creating a clear train/test split to better evaluate generalization.

This work highlights the strengths and limitations of CLIP-like models in constrained settings while paving the way for future enhancements.