Laboratory

file on

AGENTIC AI

School of Engineering and Technology

Department of Computer Science and Engineering

Subject code – **CSCR3215**

SUBMITTED BY:
Aishwarya Shelke
2023000836

SUBMITTED TO:
Mr. Ayush Kumar Singh

Sharda University

Greater Noida, Uttar Pradesh

Lab 1 - Fine-Tuning BLIP Model on an Image Captioning Dataset

Aim

To fine-tune a pre-trained BLIP (Bootstrapping Language–Image Pretraining) model on a custom image-caption dataset in order to generate accurate and meaningful captions for images.

Objective

1. To understand the working of vision–language models.
2. To use a pre-trained BLIP model for image captioning.
3. To fine-tune the BLIP model on a custom dataset.
4. To improve caption generation accuracy for domain-specific images.
5. To save and reuse the fine-tuned model.

Theory

BLIP (Bootstrapping Language–Image Pretraining) is a vision-language model developed to understand the relationship between images and text. It is trained on large-scale image-text datasets and can perform tasks such as image captioning, visual question answering, and image-text matching.

Fine-tuning BLIP involves training an already pre-trained model on a smaller, task-specific dataset. Instead of learning from scratch, the model adjusts its existing weights to better fit the new dataset. This approach reduces training time and improves performance on specific tasks.

In image captioning, the BLIP model takes an image as input and generates a textual description that accurately represents the image content.

Software Requirements

- Python 3.x
- Google Colab / Jupyter Notebook
- PyTorch
- Hugging Face Transformers Library
- PIL (Python Imaging Library)

Hardware Requirements

- System with GPU support (recommended)
- Minimum 8 GB RAM

Dataset Description

The dataset consists of:

- Images
- Corresponding captions describing each image

Each image-caption pair is used to train the BLIP model so that it learns to associate visual features with textual descriptions.

Working

Step 1: Import Required Libraries

Necessary libraries such as PyTorch, Transformers, and PIL are imported to handle deep learning operations, model loading, and image processing.

Step 2: Load Pre-trained BLIP Model and Processor

The BLIP processor converts images and text into tensors, while the BLIP model generates captions based on the given input. Using a pre-trained model helps reduce training time.

Step 3: Load and Prepare Dataset

The dataset containing image-caption pairs is loaded. Each image and its corresponding caption are read for preprocessing.

Step 4: Preprocessing

- Images are resized and normalized.
- Captions are tokenized.
- Both image and text inputs are converted into tensors using the BLIP processor.

This step ensures the data is in a format that the model can understand.

Step 5: Create DataLoader

A DataLoader is used to:

- Load data in batches
- Shuffle data
- Improve training efficiency

Step 6: Training the Model

During training:

1. The image and caption are passed to the BLIP model.
2. The model predicts captions.

3. Loss is calculated by comparing predicted captions with actual captions.
4. Backpropagation is performed to update model weights.

This process is repeated for multiple epochs.

Step 7: Optimization

An optimizer such as AdamW is used to minimize the loss and improve model accuracy.

Step 8: Save the Fine-Tuned Model

After training, the fine-tuned BLIP model and processor are saved for future use without retraining.

Result

The BLIP model is successfully fine-tuned on the custom image-caption dataset. The fine-tuned model generates more accurate and context-aware captions compared to the original pre-trained model.