AGENTIC AI LABORATORY

Presented by:

Vaidik Kumar (2024341781)

Under the supervision of:

Mr. Ayush Kumar Singh

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

SHARDA SCHOOL OF ENGINEERING AND TECHNOLOGY

GREATER NOIDA

Working of the code in the python notebook

The notebook is designed to set up the environment and data necessary to fine-tune the **BLIP** (Bootstrapping Language-Image Pre-training) model for an image captioning task. The visible cells cover the **setup** and **data loading** phases.

1. Environment Setup

The first few cells install the necessary Python libraries.

Transformers Library Installation:
Python
!pip install git+https://github.com/huggingface/transformers.git@main

- This command installs the Hugging Face transformers library directly from the main branch of its GitHub repository. This is often done to access the latest models or features (like BLIP support) that might not yet be available in the stable PyPI release.
    - *Note:* The execution logs in the notebook show an installation from a specific branch (blip-train-support), suggesting the code was previously run using a custom branch to support BLIP training before it was merged into the main library.

Datasets Library Installation:
Python
!pip install -q datasets

- This installs the datasets library, which provides utilities to easily download, cache, and process data from the Hugging Face Hub.

2. Loading the Dataset

The code loads a specific image-captioning dataset containing football images.

Loading the Data:
Python
from datasets import load_dataset

dataset = load_dataset("ybelkada/football-dataset", split="train")

-
    - load_dataset: Downloads the "ybelkada/football-dataset" from the Hugging Face Hub.
    - split="train": Specifies that the training split of the data should be loaded.

    - The output indicates the dataset is cached in the local directory (typically ~/.cache/huggingface/datasets).

3. Data Inspection (Exploratory Data Analysis)

The notebook then verifies that the data has been loaded correctly by inspecting the first example.

Retrieving a Caption:
Python
dataset[0]["text"]

- This accesses the text label (caption) of the first item in the dataset (index 0).
    - **Output:** "Benzema after Real Mardid's win against PSG" (This confirms the data includes captions describing specific football events).

Retrieving an Image:
Python
dataset[0]["image"]

- This accesses the image object of the first item. In Jupyter notebooks, running a cell with a PIL (Python Imaging Library) object as the last line automatically renders and displays the image.
    - **Output:** Displays the corresponding image of the player (Karim Benzema).

Summary of Workflow

1. **Install Dependencies:** Get the latest tools for working with vision-language models.
2. **Get Data:** Download a specialized dataset (football images).
3. **Verify Data:** Check sample text and images to ensure they match expectations before starting the training process.

Note: The provided file content ends after the data inspection step. A complete fine-tuning notebook would typically continue with:

- **Preprocessing:** converting images to tensors and tokenizing text using a BlipProcessor.
- **Model Loading:** Loading the pre-trained BlipForConditionalGeneration model.
- **Training:** Setting up a DataLoader and running a training loop (using PyTorch or the HF Trainer API) to adjust the model's weights to generate better captions for this specific dataset.