CSE 597 - Homework 1: CLIP-Based Composed Image Retrieval

Due: Sep 20, 2024 @ 11:59 pm EST

1 Introduction

For this assignment, we will focus on implementing Composed Image Retrieval (CIR). Unlike the Text-Image Retrieval task, which retrieves images according to texts, composed image retrieval aims to identify the target image corresponding to the input query, composed of a reference image and a text modifier describing how the reference image should be modified. We employ a large pre-trained vision-language model CLIP as the backbone to efficiently train a CIR model.

2 Setup

For this and the rest of the assignments, we will use Pytorch. If you are not familiar with it, there are two resources that can help you get started:

- PyTorch Official Tutorials
- Dive into Deep Learning

We suggest students complete assignments in Google Colab in a GPU environment, so you can familiarize yourself with the platform for future assignments. If you want to complete the assignments in Colab, visit the Colab website (`https://colab.research.google.com/`) and upload the assignment notebook (.ipynb). **To use a GPU, set your runtime to include a hardware accelerator first.** We suggest looking at the included software setup document (software_setup.pdf) as a general guide for working with Colab and Jupyter. The following are the steps to get the code and data ready:

1. Get the code from Canvas and unzip it. If you are using Colab, upload each file one by one to Colab (You should upload `main.ipynb` first to initialize a notebook).
2. Download the data from this link and fully unzip it (three files are zipped into one zip file).
3. Put the `data/` folder under CIR. On Colab, you can upload the data to your Google Drive and mount the data (see Colab I/O). Alternatively, you can also directly copy and create a shortcut of the shared data in your drive (see this Colab reference notebook). Remember to change the path to the data accordingly.
4. Run `python resize_images.py` to resize images. If you use Colab and have a pro account, you can use the terminal provided by Colab. If you do not have a pro account, you can use `!` in Colab to run a bash command, such as `!unzip`, `!mv`.
5. Run `pip install -r requirements.txt` to install the required Python packages.

Expected folder structure:

```
.
 data
    captions
    images
    image_splits
    resized_images
 data_loader.py
 main.ipynb
 requirements.txt
 resize_images.py
 todo.py
 utils.py
```

3 TODOs

In the CIR task, vision and text features are extracted by vision and text encoders, respectively. Then, vision and text features are fused to get fused features that are expected to be similar to the target features, so we can calculate their similarity and retrieve images according to the sort of similarity.

You are expected to implement the file `todo.py`:

- Complete the function `encode_image` (2 points)
- Complete the function `encode_text` (2 points)
- Implement your own fusion module `Combiner` (3 points)
- Complete the training loop in `train` (3 points)

Then, run `main.ipynb` to train and evaluate your model. If your fusion module is based on a basic concatenation operation and hyper-parameters are default, the expected best development score will be around 0.8-0.9. You can design your own fusion model or change the hyper-parameters to achieve higher retrieval performance.

Note: You should focus on completing the TODOs. The code will be evaluated automatically, so changing the framework of the code may lead to incorrect evaluation. If the code can't be executed there will be a **1.5-mark** deduction.

4 Submit

To submit the assignment, upload the completed `todo.py` file and screenshots of your training log. You **do not** need to include any other files (checkpoints or h5py).

5 Additional Resources

- Python and Numpy Tutorial: `https://cs231n.github.io/python-numpy-tutorial/`
- Introduction to PyTorch: `https://pytorch.org/tutorials/beginner/basics/intro.html`