README

We implemented 2 approaches for semantic search. First one is based on sentence similarity using the glove model.In which, search query and function summaries(created by transformer's model) are compared. Second approach is based on docstring vector(created by ALBERT) and function vector(learned from transformer's encoder) similarity based on greedy search on a graph data structure.

Approach 1:

1. The entire data/corpus should be saved in the "**CS657 IR PROJECT**" **Directory**. The link for the folder containing the data, csv's, jsons is given below: https://drive.google.com/drive/folders/1aa83bt3EAycvywMaa0SE47OEyEttGbP6?usp=sharing

2. Download Jsons:
    a. Download the "**jsons**" **folder** present in the link shared above and paste it in your "**CS657 IR PROJECT**" **folder** present in your google drive.

3. Download Dataset :
    a. Download the "**python**" **folder** present in the link shared above and paste it in your "**CS657 IR PROJECT**" **folder** present in your google drive.

4. To run the first part of approach one upload the "**Part 1 IR PROJECT.ipynb**" in the google colab and run it.
   Note: You will have to mount the google drive in which the "CS657 IR PROJECT" folder is present, containing the "python" folder containing json lists (jsonl's) of python code.
5. If you plan to **not** run the part 1 file of the approach 1, you can download all the resulting csv's, txt's from part 1 as follows:
    a. Download "docstring.txt", "function_tokens.txt", "train_sorted.csv" from the above link and paste it in your "**CS657 IR PROJECT**" **folder** present in your google drive.
6. To run the second part of approach one, upload the **"Part 2 Approach 1 IR PROJECT.ipynb**" in the google colab and run it.
   Note:This code will take a lot of time (4+ days) to run, we recommend using the GPU provided by colab to run the code. We ran our code in batchesin google colab pro so were able to produce results.
7. If you plan to **not** run the part 2 file of the approach 1, you can download the resulting json file as follows:
    a. Download the "**json**" **folder** present in the link mentioned above and paste it in your "**CS657 IR PROJECT**" **folder** present in your local machine.

8. To run the Flask application in your local machine, first download the "**Flask**" **folder** present in the link mentioned before and paste it in your "**CS657 IR PROJECT**" **folder** present in your local machine.

9. Before running the Flask application, install all the requirements by using the following command:
   "pip install -r requirements.txt"

10. To run the flask app open your terminal and cd to the "**Flask**" **folder** present in your "**CS657 IR PROJECT**" **folder** and run the following command:
    "python app.py"

Approach 2:

Part 1: The entire dataset should be saved in the "**CS657 IR PROJECT**" **Directory**. The link for the folder containing the data, csv's, jsons is given below:
https://drive.google.com/drive/folders/1aa83bt3EAycvywMaa0SE47OEyEttGbP6?usp=sharing

1. Download Jsons: Download the "**jsons**" **folder** present in the link shared above and paste it in your "CS657 IR PROJECT" folder.

2. Download Dataset : Download the "python" folder present in the link shared above and paste it in your "CS657 IR PROJECT" folder. To run the first part of approach one upload the "Part 1 IR PROJECT.ipynb" in the google colab and run it.

Part 2: Separated as 2.2 and 2.3 because of different tensorflow versions.

1. Run the notebooks in google colab.

2. Download 'create_pretraining_data.py', 'run_pretraining.py', albert_base_3 folder from the drive link provided. Can also download 'docstrings.txt', 'tf_examples2', 'pytorch_model.bin' to skip the time consuming steps.

3. Output is a 'embeddings.tsv' file.

Part 3: Output of part 2 i.e. 'embeddings.tsv' is needed. It is mapped to tokenized functions. Job of the transformer model is to learn mapping vectors from python functions considering docstring labels as true labels. Search string is encoded the same way as docstring(using ALBERT) and similar function vectors are fetched.

,