Directions to Navigate code:

NOTE: All the files should be run from the root directory/folder

**Config.py**

All the model configurations are present in **config.py** in the root directory. In case someone wants to test different models, MODEL\_NAME corresponding to that particular model can be uncommented.

The same can be done to change transformer embedding models.

Config.py serves as a one-stop destination to carry out the entire process with different models seamlessly.

**Scripts folder**

The scripts folder consists of all the code written to support and test the models. It also consists of the code to perform data preprocessing and indexing.

1. For data preprocessing, the **scripts/preprocess\_pmc.py** and **scripts/preprocess\_pubmed.py** files should be run individually.
2. For FAISS indexing **scripts/index\_doxuments1.py** should be run.
3. To test the code on a single customized query, **scripts/test\_query.py** should be run
4. To test model performance on the MedQA dataset, **scripts/test\_scores.py should be run.**
5. To test the model outside the RAG framework, open **scripts/test\_scores.py** or **scripts/test\_query.py** depending on the use case and comment the following line of code:

from models.generator\_LC import generate\_answer

and uncomment the following line of code:

#from models.generateNR import generate\_answer

**Models folder**

The models folder contains all the code required to set up the retriever model, generator model and to load the models inside these.

1. To make any changes/inspect how the models are being loaded open the **models/load\_model.py** file.
2. Note that the model is changed from config.py and not from the above file.
3. To inspect the retriever architecture, open the **models/retriever\_MedCPT.py** file.
4. The generator with the RAG framework is present in the **models/generator\_LC.py** file.
5. The generator without the RAG framework is present in the **models/generateNR.py** file.

The following dependencies might need to be installed to run the code.

transformers

torch

faiss-cpu

sentence-transformers

langchain

numpy

sklearn

pandas  
tqdm

matplotlib

re

Huggingface\_hub

NOTE: You will need to request access to the hugging face models used in this project to run the code. The models used are open-source, so the access is provided almost instantly. This can be followed by the **huggingface-cli login** command on the terminal and pasting your access token.