**TREC-COVID iPython Notebooks for OHSU Runs (Rounds 1-4)**

By: Jimmy Chen, OHSU MS4 (11/23/2020)

**ABOUT THE CHALLENGE**

TREC-COVID is an information retrieval (IR) challenge built on the TREC model that tasked researchers to evaluate IR systems over 5 rounds to retrieve manuscripts relevant to topics about COVID-19 from a document collection with the goal of building a reusable test collection for further research. The document collections were derived from snapshots of the COVID-19 Open Research Dataset (CORD-19), a regularly updated dataset of manuscripts consisting of coronavirus-related research gathered from various sources including journal articles, PubMed references, arXiv, medRxiv, and bioRxiv.

OHSU participated in this challenge in a team led by Jimmy Chen (OHSU SOM Class of 2021) and Dr. William Hersh, MD.

**ABOUT THIS REPOSITORY**

This repository contains the source code I used to generate runs for Rounds 1-4 during the TREC-COVID challenge. These runs consistently placed within the top 10-25 throughout these runs. I generated these runs using iPython notebooks as part of a learning process. In all of these notebooks, the base of my information retrieval system was Anserini, a open-source information retrieval toolkit.

As of 11/23/20, these notebooks cannot be run straight out of the box. They will require various files:

1. Topic file

-This is a set of query, question, and narrative for multiple topics required for a given round

-My code should output a cleaned version of the default TREC-COVID topic XML file.

2. docid file

-This is a list of valid document IDs (or docids) that will be judged during a round

3. Indexes

-I used pre-built indexes of the CORD 19 dataset, as available from the following link: [https://github.com/castorini/anserini/blob/master/docs/experiments-cord19.md#pre-built-indexes-all-versions](https://github.com/castorini/anserini/blob/master/docs/experiments-cord19.md" \l "pre-built-indexes-all-versions)

4. Qrels

-Qrels are the manually judged documents that were deemed to be relevant, partially relevant, or irrelevant to the challenge. They are used as feedback for fine-tuning systems. In my case, I tried using Qrels to fine-tune a BERT neural re-ranker, which ultimately did not outperform my Anserini system.

These files are available from: https://ir.nist.gov/covidSubmit/data.html

\*\*\*Of note, this notebook will require you to fill in the address/location of the files you downloaded from above and replace them with the addresses that contain the location of the data I used.

**INSTALLATION REQUIREMENTS**

**Will need Jupyter Notebook installed to open my notebooks. If you use Windows or Mac OS X, I recommend installing a package manager such as conda (can be Anaconda or Miniconda) and installing jupyter-notebook to run my iPython notebooks. My code was also developed on Python 3.6.**

**Pre-requisite packages:**

1. Pyserini (pip install pyserini)

2. json (think it’s built in)

3. Pandas, numpy, os (should be installed with your python environment)

4. TrecTools (pip install trectools – mandatory to run trec\_eval on your runs)

4.Optionally, if you want to run the NLP stuff (paragraph tokenization and ranking), you will need the following:

-transformers, tensorflow, keras, nltk

**Setting up Pyjnius (Required):**

Because Lucene was built using java, we need a python package to bridge Java -> Python). Pyjnius fits this need. This package should be installed with Pyserini when you run the above command, but you will need to first, install a Java Development Kit (JDK) and second, set your environmental variables to point to the java virtual machine DLL (jvm.dll) to use Pyjnius.

This thread shows how to do set up your environmental variables on Windows: <https://stackoverflow.com/questions/20970732/jnius-1-1-import-error>. This is what I used

This thread shows how to do it on Mac OS: <https://pyjnius.readthedocs.io/en/stable/installation.html> and <https://stackoverflow.com/questions/22842743/how-to-set-java-home-environment-variable-on-mac-os-x-10-9>