Cody Webster

Progress Report

1) Progress made thus far

   Generic

   Wrote script(s) to process documents and convert to a json format for easy loading in the future. Json files contain tokenized representations of the text and queries

   Cranfield

   Wrote script to convert json text data into Cranfield datasets as required for the metapy module.
   Adapted the scripts for homework to find the optimal values of weights in for the JM and BM25 algorithms.
   Adapted the scripts for homework to ranking documents.

   BERT

   Downloaded Google's MS-Marco pre-trained BERT model.
   Wrote script to examine documents and determine missing vocab that is common to the corpus.
   Adapted the scripts in the git repo (https://github.com/cognitiveailab/ranking) to finetune and train the model with the tensorflow_ranking python module.
   Adapted the scripts in the aforementioned git repo to run the scripts in a Docker container running in Window's Subsystem for Linux on my desktop's GPU.
   Rewrote the script for running to Docker to run outside of Docker for use with Google COLAB.
   Wrote script to convert ranking output into a predictions file.

2) Remaining Tasks

   Generic

   Consolidate scripts and files that are spread over several directories and add into the git repo.

   Cranfield

   Run again with titles included in the "text" (not likely to beat baseline)

   BERT

   Finish running the BERT model and compile results. Each document/query combination that is longer than 512 broken tokens has been into segments. I will test whether a mean, geometric mean, or max score within the segment is a better metric.
   Test if I can use a larger BERT model with Google Colab (currently using BERT Small)

3) Any challenges
   - Adapting the examples in the git repo so that I can run the BERT model on my system or in git
   - Hardware limitations on my PC. Partially solved by using Google COLAB
   - Scoring the documents with BERT takes an excessive amount of time due to the large number of documents in the corpus and the limit of 512 tokens per analysis run. This excessive wait on results has seriously slowed my progress.
   - Actually beating the baseline performance.