Team: NA coders

Members: Hoa Le (NetID: hle30), Nathaniel Rupsis (NetID: nrupsis2)

Project Progress Report

For the project our team decided to participate in the CORD-19 Open Research Dataset IR Competition. The CORD-19 dataset contains over 57000 scholarly articles available for the global research community. Our goal is to build a live system that supports search on this massive dataset.

Progress Made Thus Far

We break down the project into smaller tasks and milestones: extracting, preprocessing, indexing, retrieving and ranking.

In the extracting step, we collected Titles, Abstracts, and Introductions of papers using the Python code sample provided. Then, we fed the extracted data to the preprocess function we created. The preprocess function contains basic preprocessing tasks such as removing special characters, text.split(), stemming, etc…

For data indexing, we've taken two different approaches, and each individual is working on their own implementation. By doing this, we'll have some options and flexibility when tweaking the overall system.

The first approach is to build inverted indices from scratch. It will take 4 inputs (uid, title, abstract and keywords) or a dataframe containing those 4 elements and output a dictionary of inverted indices. These keywords are based on their tf-idf scores. We used the stop word file for previous assignments to obtain a list of words that will not be indexed. As for the query, we decided to go for the query field of the topic. They are easily preprocessed as the other two fields(question and narrative) contain uppercase and more special characters.

The Second approach is to build a corpus .dat file with a pre-processor, and then utilizing the metapy indexing algorithm. By doing this, it allows us to focus on what to include in the index (title, author, etc), while freeing us from having to worry about the performance of the index.

With both index approaches, our team has a firm handle on the data preparation, and the next task is to implement the retrieval model. For the ranking algorithm, we are looking to use okapi BM25 and combine it with other state-of-the-art methods.

Remaining Tasks

Even though we haven't gotten results from our okapi BM25 ranking algorithm, we are 90% sure that it won't be enough to beat the baseline. Thus, the remaining tasks would be to finalize all the steps we mentioned, testing BM 25, fine-tune various parameter settings and then look to combine it with different methods to enhance our result. We are currently looking at some of the live systems that use BM25 such as Neural Covidex. We are also exploring LDA(Latent Dirichlet Allocation) which studies semantic relationships.

Challenges that We Encountered

We had trouble getting started because we felt overwhelmed with the massive dataset which contains over 57000 articles. However, once we started to break down the project into smaller tasks, it became much more manageable. The preprocessing proved to be a bit of a tough task since the dataset contains highly technical papers with scientific terms. Another big challenge is to build an inverted index from scratch. I think it's extremely worthwhile and we can learn a lot from the process. Final challenge to find the advanced ranking methods to pass the baseline.