CS410 Text Information Systems – Fall 2023
Project Progress Report

Gilberto Ramirez
ger6@illinois.edu

November 6, 2023

1 Which tasks have been completed?

I managed to complete the following tasks:

- *(6 hours) Fetch and filter corpus.* I created a Python program, `get_rfcs.py`, which makes use of the `rsync` service provided by the RFC editor website to retrieve the 10,000+ RFCs and related documents the first time and those added/updated/deleted in subsequent calls. In addition, the program creates the index file, `rfcs-full-corpus.txt`, containing only those files (RFCs) needed to be part of the inverted index and needed by MeTA to access the corpus. Finally, the program deletes and recreates the inverted index if the corpus changes.
- *(12 hours) Python back-end for information retrieval.* I coded and tested the Python back-end program, `search.py`, that uses the inverted index created by `get_rfcs.py`, and uses a BM25 (MeTA) ranker to return a list of relevant documents closest to an input document (query). Input query comes in the form of a string and output is in the form of an array of dictionaries where each array element corresponds to a relevant document returned by the BM25 ranker. The array elements are sorted in descending order by the respective document ranking score.
- *(6 hours) Python back-end application using Flask.* I created a basic application in Python, `rfc_finder.py`, using Flask that acts as an API endpoint and calls rest of backend modules. However, the `MeTA` ranker functions could not run with Flask for some unknown reason. As a result, I had to switch to another Python web-framework called Bottle to get the application working as expected.

2 Which tasks are pending?

The following tasks are pending:

- *(10 hours) Chrome extension creation* Learn about developing Chrome extensions and code the front-end using JavaScript, HTML, and CSS.
- *(2 hours) Testing.* Test the solution and adjust ranking function or inverted index creation if necessary.
- *(4 hours) Documentation.*
- *(5 hours) Project report.*
- *(3 hours) Project presentation.*
- *(Optional: 20 hours) Topic Mining.* Discover topics in the corpus using LDA and allow Chrome extension to list topics discovered in RFC opened in the active browser tab (if any).

3 Are you facing any challenges?

I spent a lot of time trying to make work Flask with MeTA but was not successful and, in the end, switched to another web-framework called Bottle. Also exploring how to create topics using LDA and trying to figure out how that works in MeTA and best parameters including number of topics, `alpha` value, `beta` value, and algorithm (CVB0, Parallel Gibbs...). Unfortunately MeTA website seems to be inaccessible due to the expiration of the website certificate. I reported the issue in Campuswire (posts #889 and #877).