CS410 Text Information Systems – Fall 2023
Project Proposal

Gilberto Ramirez
ger6@illinois.edu

October 25, 2023

1 What are the names and NetIDs of all your team members? Who is the captain?

Gilberto Ramirez, NetID: `ger6` – Captain and the only team member!

2 What topic have you chosen? Why is it a problem? How does it relate to the theme and to the class?

The huge success of Internet since its birthday in 1983 is largely due to a well defined set of rules of engagement that all devices need to follow when they connect in a network. These rules are formally defined in normative specifications of technologies or methodologies known as Internet Standards created and published by the Internet Engineering Task Force (IETF). Every single contribution to the IETF in the form of a memorandum is born as an Internet Draft and, after several revisions, might be accepted and published as a Request for Comments (RFC) and labeled a Proposed Standard. Later, an RFC can be elevated to a Standard when maturity has reached an acceptable level. Since the first RFC (RFC 1) written in 1969, more than nine thousand individual documents have been posted by the RFC Editor. Each RFC is submitted as plain ASCII and published in that format.

One of the major challenges for Internet standard researchers and implementers is to navigate more than 50 years of documentation in an efficient way. Rather than *googling* computer networking terms when doing research, a search using an information retrieval function like BM25 instead of exact keyword matching *limited to the IETF RFCs corpus only* might provide a more complete list of authoritative documents closely related to an input query. This approach can save users considerable time in curating the relevant RFCs that need to be studied in detail to make progress in their research tasks.

In this project, I intend to create the information retrieval system described above. Since all RFCs are hosted in the RFC Editor web site and are available in web format (HTML/CSS), I am planning to implement the system as a Chrome browser extension as detailed in the *Intelligent Browsing* theme described in the CS410 Project Topics document. This will make the tool easy to use by the researchers who normally access RFCs content using a web browser.

Since this project plans to implement BM25 to solve an information retrieval problem of the IETF RFCs corpus, it will require a thorough understanding of several of the following class lectures:

- Text Retrieval Problem,
- Text Retrieval Methods,
- Vector Space Model including the BM25 Scoring Function, and
- System Implementation including Crawling and Inverse Index.

3 Briefly describe any datasets, algorithms or techniques you plan to use

The main dataset (corpus) is all the IETF RFCs hosted in the RFC Editor web site. Despite having around 10,000 documents in this corpus, they are all in ASCII format and do not change too frequently. I am planning to keep all the RFCs in local storage in order to create an inverted index that might need to be refreshed once a week. I will use this inverted index to run the BM25 ranking function given its flexibility.

If time allows it, I might add functionality to discover topics for an RFC chosen by the user, i.e., a user with a browser tab opened with a specific RFC document in the RFC Editor web site can click on the extension icon to list topics that RFC touches and discovered in an unsupervised way, so the researcher can fetch a quick summary of the main areas the RFC covers before embarking in the long journey of detailed reading and analysis that can take many hours. For topic analysis, I am planning to use Latent Dirichlet Allocation as the algorithm of choice.

4 How will you demonstrate that your approach will work as expected

There are no formal Cranfield datasets for this problem and there are no applications where an indirect evaluation might help in the evaluation. I will run queries and evaluate, manually, the results. I will also recruit the help of researcher colleagues to provide feedback but, due to the very limited time, it might not be possible to collect sizable feedback by the time the tool is ready.

5 Which programming language do you plan to use?

I am planning to build an extension only for Chrome since that is the web browser used by most of my fellow researchers in the area. As a result, I will use JavaScript, HTML, and CSS for the front-end. For the back-end which includes crawling and scraping the corpus, creating the inverted index, and implementing BM25, I am planning to use Python, MetaPy/MeTA, and Flask.

6 Justify that the workload of your topic is at least 20 hours. List tasks to be completed and estimated time for each task.

A rough estimate of the project tasks and time I estimated I needed to complete each task:

- (4 hours) Fetch and filter corpus files. This might be making use of the `rsync` service hosted by RFC Editor. Create an inverted index using Python and MetaPy/MeTA.
- (10 hours) Learn about developing Chrome extensions and code the front-end using JavaScript, HTML, and CSS.
- (12 hours) Code the back-end in Python to be able to receive a REST based JSON request with the query keywords provided by the user and collected by the front-end, apply a ranking function (BM25) using the previously built inverted index, and return the list of documents (RFCs) in descending order of relevance.
- (2 hours) Testing the solution to adjust ranking function or inverted index creation.
- (4 hours) Documentation.
- (5 hours) Project report.
- (3 hours) Project presentation.