CS 410: Project Proposal

Free Topics: Topic querying on UIUC Slack and Campuswire

1. What are the names and NetIDs of all your team members? Who is the captain? The captain will have more administrative duties than team members.

Hung Nguyen - hungn2 [Captain]
Blake Jones - blakeaj2

2. What is your free topic? Please give a detailed description. What is the task? Why is it important or interesting? What is your planned approach? What tools, systems or datasets are involved? What is the expected outcome? How are you going to evaluate your work?

Free Topics: Topic querying on UIUC Slack and Campuswire.

The UIUC MCS Slack has a variety of course specific channels. These channels are used year over year, resulting in a buildup of relevant information for a given course. This information can be vital to students in future semesters. Campuswire is another great resource for information on the content of a course's current semester, however students must be invited to a course, and only have access to that semester's content. As a result, Campuswire does not have the rich historical history as Slack does. This disconnect between these two data sources inhibits current students from leveraging previous students' experiences in learning new content. In addition, students are inundated with a variety of posts, many of which are duplicated (e.g. a slack question that was also posted to Campuswire). Our project will aim to merge historical Slack and semesterly Campuswire whilst removing duplicated answers, providing an efficient and concise querying service for UIUC course questions.

Approach
Our CLI tool will provide three modes of querying data:

- Given an ad-hoc query, both Slack and Campuswire will be searched for relevant conversations.
- Given a Campuswire question, Slack will be searched for relevant conversations.
- Given a Slack message, Campuswire will be searched for relevant conversations.

Users will execute our CLI tool inputting an ad-hoc query, or a slack/campuswire conversation. The CLI tool will then search the appropriate data sources and return the most relevant documents to the users query, ranked by relevance.

Datasets:

- CS410 2021FA Campuswire
- #cs-410-text-info-syst Slack Channel

Tools/Systems:

- Python Click (CLI tooling)
- MetaPy
- Slack API
- Beautiful Soup (for scraping Campuswire)
- Scikit-learn (deduplication via clustering)

Expected Outcome
After inputting a query or slack/campuswire question, users can expect a ranked set of relevant documents from Slack/Campuswire. This document set will be deduplicated to minimize effort on behalf of the user.

How are you going to evaluate your work?
We will be using the Cranfield Evaluation Methodology to measure how well our system's result matches the ideal ranked list. We will be building our own test collection by running through several queries and then evaluating each document/conversation as relevant or non-relevant.

4. Which programming language do you plan to use?

Python, as we will be building a CLI based Slack/Campuswire querier.

5. Please justify that the workload of your topic is at least 20*N hours, N being the total number of students in your team. You may list the main tasks to be completed, and the estimated time cost for each task.

- [3hr] Implement CLI interface
- [9hr] Implement Campuswire library
    - Campuswire has no API, so we will create our own scraping library
- [6hr] Implement document ranking on merged Slack/Campuswire topics
- [6hr] Implemented clustering deduplication on results.
- [12hr] Creating evaluation test collection.
- [6hr] Parameter tuning
    - Clustering algorithm/parameters
    - Ranking function/parameters