**CS 410 Project Proposal:**

**Chrome Extension for Linking User Queries to Coursera Video Segments**

*Group Members for this project include:*

***Albert Sadiku*** *(NetID: asadik3, Team Captain) and* ***Mathew Tang*** *(NetID: mt13)*

For our project, we decided to choose the Intelligent Browsing topic because we wanted to provide additional functionality on top of existing web browsers. In particular, we wanted to build a Chrome extension for linking user queries to Coursera video segments. This is a problem currently because Coursera only allows you to search for exact terms when looking for a particular topic video. However, we would like to improve this by using the BM25 retrieval function on transcript data to allow for more complex searches. Instead of looking for exact search results, our extension would look at all the possible video segments (of CS410 lecture videos) and rank them in order of relevance based on the BM25 retrieval function. Then, we would show the user the most relevant video segments based on their query. This relates to the theme of Intelligent Browsing because we are improving the capabilities of the web browser via our extension. Moreover, this project relates to the class because we are using the skills and methods learned during the first half of the course—namely, document ranking and the BM25 retrieval function.

To obtain the data we need for this extension, we would scrape the lecture transcript data for all of CS 410. After scraping all the data, we would need to process it and make use of the BM25 retrieval function. To demonstrate how our algorithm will work as expected, we could do example queries and see how well it is working through user input. For example, if we are able to search a term, it could return direct users to relevant transcript locations. For the scraping portion, preprocessing portion, and BM25 portion we would use Python. We would use Javascript to develop the front end for the actual Chrome extension.

Workload of the project:

Scraping all Coursera Lectures: 10 hours

Pre-processing all data for use: 5 hours

Implementing BM25: 5 hours

Creating front end: 5 hours

Integrate front end with backend: 7 hours

Full documentation: 2 hours

Testing for functionality: 3 hours

Report: 3 hours

Total: 20 \* 2 students = 40 hours