**CS 410 FA20 - Project Progress Report**
2.2 ExpertSearch System - *Extracting relevant information from faculty bios*
Keon Park, keonp2@illinois.edu

## Background / Overview

My project is a system based on *ExpertSearch* that, when finished, will automatically scrape and parse UIUC faculty pages from user-provided URLs to extract key information about faculty members. My original proposal is available here: https://github.com/Parkkeo1/CourseProject**.**

## Progress So Far

My work on the project so far has focused on text data retrieval, processing, and some keyword extraction and topic mining. My code is still in a WIP Jupyter Notebook and has not yet been integrated into a full application.

To get started, I am currently using my dataset of UIUC CHBE faculty text data from my MP2.1. Using this dataset, I have built a web scraper that scrapes text data from HTML tags whose CSS classes match common faculty profile sections (i.e. scrape only relevant content from the page). Because faculty pages for a given department have the same or very similar format, I have implemented the ability to customize the web scraper's list of key CSS classes to match for each department at UIUC (currently only have one, for CHBE).

Using the basic CHBE faculty data, I have also developed working implementations of the following three key components in my application:

1. Tokenizer, using spaCy, with stop words + punctuation filtering and lemmatization.
2. TF-IDF weighting, using scikit-learn; the trained vocabulary/index is able to be saved and loaded for use on new faculty data.
3. LDA, using scikit-learn, the topic distributions are calculated using the training data, and the topic coverage for a new doc can be calculated too.

With these parts working, I am able to scrape relevant text data from a faculty member page URL not in the training data, tokenize/clean it, and add it to the existing TF-IDF index/vocabulary to calculate its top weighted keywords and its topic coverage using LDA.

## Remaining Tasks

1. Refactor the existing components: scraper, tokenizer, TF-IDF, and LDA to work in a smooth pipeline for both training data and new user-input data such that:
    a. Refine the scraper's ability to match relevant HTML tags for more different formats of faculty pages than just CHBE's.
    b. For new faculty's text data, use spaCy to find named entities and search for email and phone number tokens during the tokenization step,

2. Test out the above functionality with a much bigger dataset (UIUC faculty bios from MP2.3) and refine/fix things as needed (the final application's TF-IDF weights should be based on this as it should be more comprehensive over more different departments.
3. Build out final backend and frontend to package work into a full application.

**Challenges**

The main challenge I foresee in finishing this project is the frontend/UI and how I should approach designing and implementing the data visualizations for the faculty text mining results. Given that I have already put in a lot of focus and effort on the text retrieval and mining components of the project, I may have settle for more basic data visualizations in the frontend than I originally planned, in order to keep my total project development time reasonable.