

Proposal for Professors' Finder

Providing UIUC Professors' Profiles Based on Specific Topics

1. Team Members

Yi Yang (Captain) - NetID: yy52

Jeff Sun - NetID: jifeis2

2. Project Description

Our project, Professors' Finder, aims to provide a user-friendly platform where students can easily discover and access UIUC professors' profiles based on specific topics. The task involves developing a search engine that allows users to input subjects or research areas, and in return, displays a list of professors whose expertise aligns with the given criteria (the number of published paper or related courses). This project is meaningful as it streamlines the process of finding suitable mentors, research advisors, or collaborators within the university community. We plan to achieve this by crawling professors or courses page and implementing a robust search algorithm.

Planned Approach

- Utilize web scraping techniques to collect data from UIUC official websites and professors' personal websites.
- Clean the data and create dataset, each professor's data will be in a dedicated text file and the data at least include professor's name, title, research interests, course taught, public papers, etc.
- Build index with metapy.
- Implement a TF-IDF model and/or BM25 model in index query which focus on matching user-inputted topics with professors' research interests.
- Design an intuitive and responsive user interface for smooth user experience.

Frameworks

BeautifulSoup for web scraping, Express for backend development, FastAPI for building web.

Datasets: UIUC official websites and professors' personal websites.

Expected Outcome

The expected outcome is a fully functional web application, on which users can search for professors based on specific topics, view detailed profiles, and make informed decisions regarding academic and research collaborations.

Evaluation

Test the accuracy of topic matching by comparing search results with professors' actual research interests.

3. Programming Language

We plan to use Python for web scraping and web API development.

4. Justification of Workload

The workload of our topic exceeds 40 hours due to the complexity of tasks involved:

- **Data Collection:** Web scraping and cleaning the data from multiple sources (15 hours).
- **Algorithm Development:** Implementing a robust search algorithm with TF-IDF and/or BM25 (choose a better one after evaluation) for accurate topic matching (10-15 hours).
- **User Interface development:** Developing an intuitive and visually appealing frontend for the application (10-15 hours).
- **Testing and Optimization:** Rigorous testing, bug fixing, and system optimization (10-15 hours).

5. Reference and Improvement

Our topic is similar to Expert Finder [1]. The improvements and innovations in our project include:

- **Sorting Algorithm:** The previous project sorted based on query term count/biography word count. In contrast, our algorithm not only searches for query terms on personal profiles but also explores semantically similar terms using the BM25 model, taking into account not only term frequency but also relevance. Furthermore, we can consider limiting the counting scope to courses and published papers.
- **User Interface:** Our interface will be more user-friendly. Unlike the previous project, which presented search results as a list of URLs, we will display professors' photos, names, and brief introductions. Users can click on "Learn More" to navigate to the individual's profile page.

[1] Sanavaitis, J. Expert Finder. https://mediaspace.illinois.edu/media/t/0_ox09r37n/112201961, 2019.