

Advisor Matching System Project Documentation

Team Members: gnp4, nikitha6, chawhan2

1. Overview of the Function of the Code:

The Advisor Matching System is designed to automate the process of matching university students with potential advisors based on shared research interests. The system takes advantage of web scraping to gather data on professors, which includes names, titles, website links, images, research papers, and research interests. Currently, the system supports the University of Illinois Urbana-Champaign (UIUC) Computer Science faculty list, with plans for expansion to include other universities.

The core functionality involves a user input interface where students provide their research interests in keywords or sentences and/or their research paper/project titles and snippets of explanations. The system then employs similarity functions to identify and recommend suitable advisors from the scraped dataset. Two key algorithms, TF-IDF and BM25, are used to determine the best matches. The project is implemented using Python for the backend, with a React-based user interface, and Flask is utilized for seamless integration.

2. Documentation of How the Software is Implemented:

- Web Scraping:
 - We identified the UIUC CS faculty list as the target source for professor data. This typically involved locating the specific webpage where the relevant information is hosted, i.e.
<https://cs.illinois.edu/about/people/all-faculty>.
 - We then extracted relevant information such as name, image, research papers, website link, and their respective research interests from the UIUC CS faculty webpage and Google Scholar.
- User Interface (UI):
 - Developed a user-friendly UI using React JS for easy interaction with the system.
 - Integrated the UI seamlessly with the backend using Flask, ensuring smooth communication.
- Similarity Functions:
 - Employed clustering based on research interests to group professors.

- Implemented TF-IDF and BM25 algorithms to determine the similarity between user input and professors' research interests.
 - Established a ranking system to present top-matched professors to the user.
- Backend Script (Python):
 - Implemented the core functionality in Python.
 - Ensured modularity and flexibility for future enhancements.
 - Integrated the similarity functions with the UI for a cohesive user experience.

3. Documentation of Usage of the Software:

- User Input:
 - Users input their research interests through the UI, providing keywords or sentences that reflect their areas of interest.
- Matching Process:
 - The system processes user input using TF-IDF and BM25 algorithms.
 - Professors with the highest similarity scores are identified and presented as potential advisors.
- Results Display:
 - The system displays matched professors along with relevant details, such as names, images, and links to their webpage.
 - Users can explore further details and make informed decisions about potential advisors.

4. Brief Description of Contribution of Each Team Member:

- Scraping Team (gnp4, nikitha6):
 - Responsible for collecting professor data through web scraping.
 - Ensured the extraction of essential information from the UIUC faculty list.
- UI Development Team (chawhan2, gnp4):
 - Created an intuitive and visually appealing user interface using React.
 - Ensured seamless integration with the backend for a cohesive user experience.
- Backend Development Team (nikitha6, gnp4):
 - Implemented the core functionality in Python, handling user input and similarity calculations.
 - Ensured modularity and scalability for future expansions.
- Algorithm Team (gnp4, nikitha6, chawhan2):

- Developed and fine-tuned the TF-IDF and BM25 algorithms for optimal professor matching.
- Conducted clustering based on research interests to enhance the accuracy of matches.