

# Final Report For Professor's Finder

## Providing UIUC Professor's Profiles Based on Specific Topics

### 1. Introduction

Our project, Professor's Finder, is designed as an intuitive platform enabling students to conveniently engage with UIUC professors' profiles relevant to research interests they are interested in. This initiative involves developing a search engine where users can enter specific subjects or research areas. The engine then showcases a list of professors whose expertise and academic contributions, like research areas and recently taught courses, match the search query.

This project holds great value since it simplifies the process of identifying appropriate mentors, research advisors, or collaborators within the academic community. Our approach includes web scraping to gather information from professors' and courses' pages and the development of an efficient search algorithm to ensure accurate and relevant results. The ultimate goal is to enhance the academic journey of students by providing them with a streamlined tool for academic networking and mentorship opportunities.

### 2. Methodology

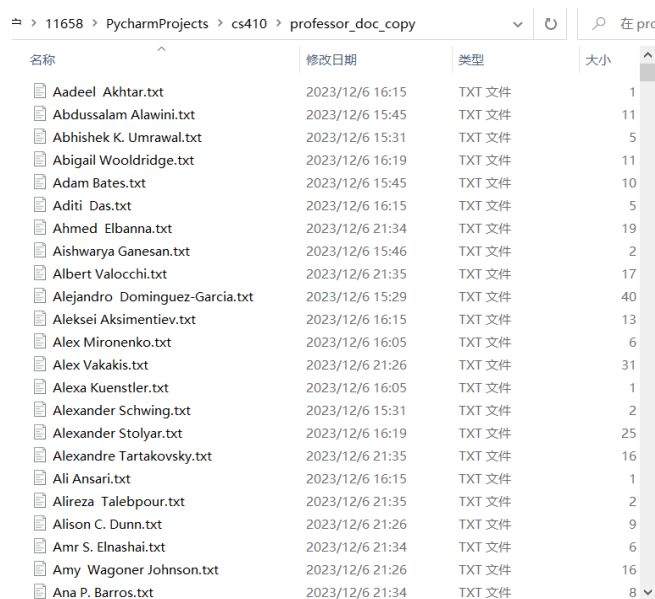
#### 2.1 Web Crawling

We used webdriver to automatically control browser actions and parse HTML documents for data extraction using BeautifulSoup. We scraped faculty information from the official

```
{
  "name": "Jonathan Freund",
  "department": "aerospace",
  "photo_url": "https://ws.engr.illinois.edu/directory/viewphoto.aspx?id=7726&s=400&type=portrait",
  "home_page_url": "https://aerospace.illinois.edu/directory/profile/jbfreund",
  "doc_name": "jonathan-freund",
  "brief": {
    "title": "Willett Professor, Department Head",
    "research_area": [
      "Aeroacoustics",
      "Combustion and Propulsion",
      "Computational Fluid Mechanics"
    ],
    "courses_taught": [
      "AE 403 - Spacecraft Attitude Control",
      "AE 590 AO (AE 590 ONL) - Seminar",
      "AE 598 ONL (AE 598 UQ) - Uncertainty Quantification",
      "AE 599 JBF - Thesis Research",
      "ME 410 - Intermediate Gas Dynamics",
      "TAM 335 AL1 - Introductory Fluid Mechanics"
    ]
  }
},
```

Fig1. Professor index

websites of nine departments within the Grainger college of engineering, resulting in two sets of output files. One is an index json file, in Figure1 containing brief information that will be displayed for each professor on the webpage, including name, department, title, photo URL, homepage URL, taught courses and research areas. Another set of documents are txt files for each professor, containing their personal information in far more details such as biography and published papers, as Figure 2.



名称	修改日期	类型	大小
Aadeel Akhtar.txt	2023/12/6 16:15	TXT 文件	1
Abdussalam Alawini.txt	2023/12/6 15:45	TXT 文件	11
Abhishek K. Umrawal.txt	2023/12/6 15:31	TXT 文件	5
Abigail Wooldridge.txt	2023/12/6 16:19	TXT 文件	11
Adam Bates.txt	2023/12/6 15:45	TXT 文件	10
Aditi Das.txt	2023/12/6 16:15	TXT 文件	5
Ahmed Elbanna.txt	2023/12/6 21:34	TXT 文件	19
Aishwarya Ganesan.txt	2023/12/6 15:46	TXT 文件	2
Albert Valocchi.txt	2023/12/6 21:35	TXT 文件	17
Alejandro Dominguez-Garcia.txt	2023/12/6 15:29	TXT 文件	40
Aleksei Aksimentiev.txt	2023/12/6 16:15	TXT 文件	13
Alex Mironenko.txt	2023/12/6 16:05	TXT 文件	6
Alex Vakakis.txt	2023/12/6 21:26	TXT 文件	31
Alexa Kuenstler.txt	2023/12/6 16:05	TXT 文件	1
Alexander Schwing.txt	2023/12/6 15:31	TXT 文件	2
Alexander Stolyar.txt	2023/12/6 16:19	TXT 文件	25
Alexandre Tartakovsky.txt	2023/12/6 21:35	TXT 文件	16
Ali Ansari.txt	2023/12/6 16:15	TXT 文件	1
Alireza Talebpour.txt	2023/12/6 21:35	TXT 文件	2
Alison C. Dunn.txt	2023/12/6 21:26	TXT 文件	9
Amr S. Elnashai.txt	2023/12/6 21:34	TXT 文件	6
Amy Wagoner Johnson.txt	2023/12/6 21:26	TXT 文件	16
Ana P. Barros.txt	2023/12/6 21:34	TXT 文件	8

**Fig2. Professor information documents**

## 2.2 Search Algorithm

### Use BM25 with rank\_bm25.BM25Okapi to compute score of relevant documents

In Professor Finder we implemented the BM25 algorithm to rank a professor's relevant document. The BM25 algorithm is a widely-used ranking function used by search engines to estimate the relevance of documents to a given search query. It's based on the probabilistic information retrieval model and is a variation of the TF-IDF (Term Frequency-Inverse Document Frequency) approach. We use BM25Okapi class in package rank\_bm25 which allows us to easily compute bm25 scores of a collection of documents. In BM25 we chose parameters  $k1=1.5$  and  $b=0.75$  to try to get a more reasonable score for the professor's relevant documents.

### Support Keywords Stem Search

In our project, we implemented a keyword stem search feature using the nltk library, specifically leveraging its "punct" tokenizer. This functionality will enhance search accuracy and efficiency by identifying and matching word stems, improving the overall user experience.

## Support Keywords Synonyms Search

We also implemented a keyword synonym search capability with the nltk library, making use of wordnet. It will enhance search results by recognizing synonyms, expanding search queries, and delivering more comprehensive and relevant information to users. This feature can improve user satisfaction and search precision.

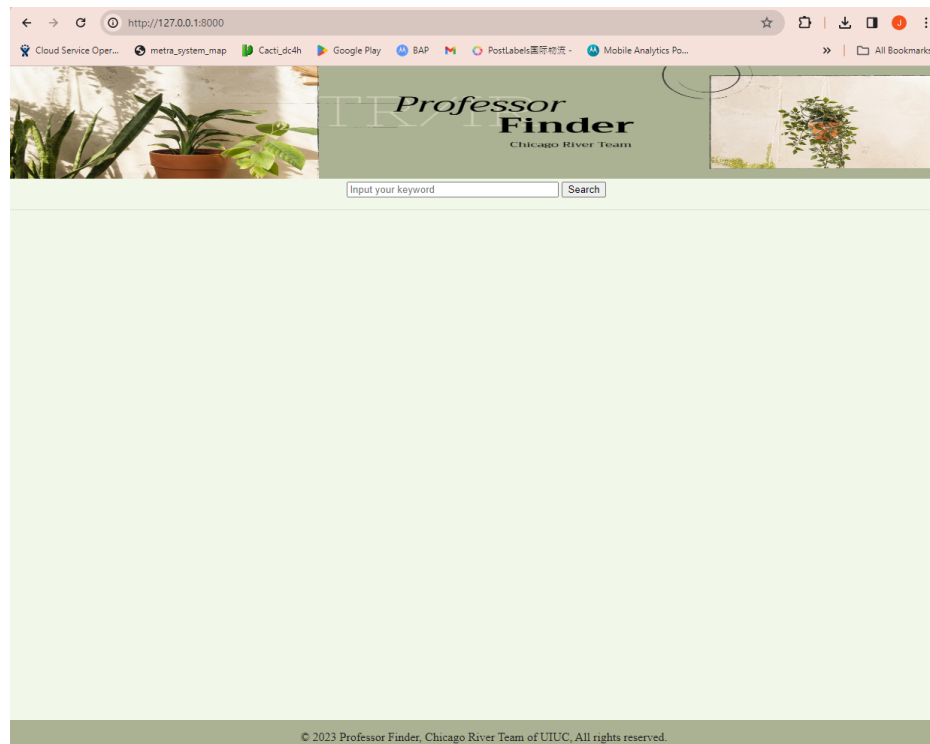
## Support Case Non-Sensitive Keyword Search

The website also supports case non-sensitive keyword search. The search engine treats uppercase and lowercase letters as equivalent, so users need not worry about letter casing when they enter search queries. We achieve this feature by turning the keywords and all documents into lower case before computing. It improves user-friendliness and search convenience of our platform and delivers more accurate search results.

## 2.3 Front-end Development

We employed the FastAPI along with the unicorn framework to develop the front-end of our website. FastAPI is a modern, fast and light web framework for Python. The programming languages used for front-end development include Python, JavaScript, HTML and CSS. This combination allowed us to create a robust, interactive, and user-friendly website.

## 3. Results



**Fig3. Homepage of the website**

Figure 3 shows the homepage of the website. In the search box, users can enter queries such as specific research areas or majors.

Figure 4 displays the search results for "network," sorted by relevance from high to low. Scrolling and turning page can provide additional search results. Users are able to view the professor's introduction on the right-hand side of the interface and click on the professor's photo or name to navigate to their individual web page for additional information.

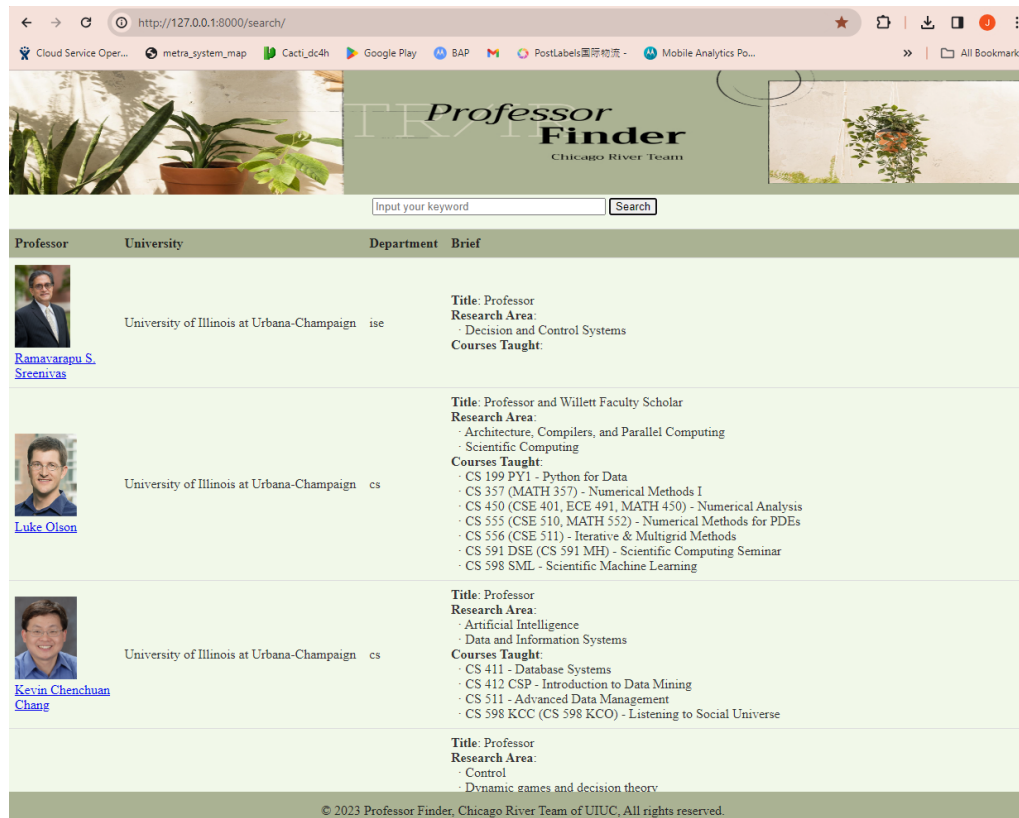


Fig4. Search results for “network”

## Keywords Synonyms Search

From the backend, a list of synonyms derived from the search term can be observed. This increases search robustness by expanding beyond a single query term. For example, entering “networking” will display search results for both “network” and “networking” as Figure 5.

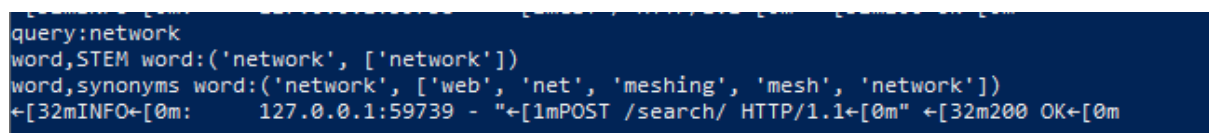


Fig5. Backend message for searching “network”

## Keywords Stem Search

As Figure 6 shows, when users enter “information” as a query term, inform is listed as stem word. It intensifies search efficiency along with the feature of keywords synonym searching.

```

[32mINFO[0m:      Started server process [36m1740[0m]
[32mINFO[0m:      Waiting for application startup.
[32mINFO[0m:      Application startup complete.
query:information
word,STEM word:('information', ['inform'])
word,synonyms word:('information', ['information', 'selective', 'information', 'entropy', 'data', 'info', 'inform'])
[32mINFO[0m:      127.0.0.1:58502 - "[1mPOST /search/ HTTP/1.1[0m" [32m200 OK[0m

```

**Fig6. Backend message for searching “information”**

## 4. Challenges and Solutions

### 4.1 Compatibility Challenges

The official websites of various departments within the Grainger College of Engineering share a similar structure and layout, but there are still differences. For instance, the URL for the ECE department's faculty page is “<http://ece.illinois.edu/about/directory/faculty-dept>”, while the CS department's URL is “<http://cs.illinois.edu/about/people/all-faculty/department-faculty>”. Additionally, some websites display four professors in a row, while others have three or five. To accommodate these variances, we have implemented adaptive measures such as pre-determining the department or pre-reading the HTML tags to increase compatibility.

### 4.2 Search Optimization

To improve search result optimization, we should implement beyond to match exact query terms. We applied the nltk library to support keyword synonym search and stem search. Matching search query terms with synonyms or closely related words to offer more comprehensive and dependable results.

## 5. Self-Evaluation

Initially, 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. At medium state, we planned to implement keyword synonym search and stem search and improve the user interface. Finally, at the current state, we have implemented all these issues.

Our idea is inspired by the Expert Finder project [1]. The improvements and innovations in our project include sorting algorithm, keyword synonym search and keyword stem search. 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, our interface is more user-friendly. Unlike the previous project, which presented search results as a list of URLs, we display professors'

photos, names, and brief introductions. Users can click on photos or names to navigate to the individual's profile page.

## **6. Future Work**

Currently, our web scraping script is tailored to work exclusively with the websites of the nine departments under the College of Engineering. Expanding compatibility to more departments or universities can be a remaining task. Additionally, we can explore the complexity of automatically rectifying typographical errors in order to enhance user query convenience and intelligence.

## **7. Conclusion**

We've created a valuable tool for academic networking by utilizing web crawling, BM25 ranking algorithm and user-friendly interface design. Despite challenges in compatibility and search optimization, our adaptive solutions have ensured a convenient and efficient user experience. Looking ahead, expanding this platform to include more departments and enhancing query intelligence remain exciting prospects. This project not only serves UIUC's immediate community but also sets a precedent for academic search tools in higher education.

## **8. Reference**

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