UIUC Expert Search Engine

Alvin Jou

Abstract— **The UIUC Expert Search Engine is a search engine that allows for an easy way to find research experts at UIUC in a particular field. This paper will go over my process for creating the search engine.**

I. INTRODUCTION

Currently the best way to find experts in a particular field at UIUC is to simply Google. Relying on Google may not necessarily be the most effective way to approach this problem though. Even after you get results from Google, you need to parse through the results to find out the actual names of the experts. Then you need to do another search to find the contact information about that professor. The UIUC Expert Search Engine will simplify this process, and simply return the relevant information about the professor directly.

The novelty of this project involves that fact that in encompasses all of the faculty at UIUC, instead of just a particular college. The project is built upon an ElasticSearch index, which we populated with information from web crawling. After the index is built, some experimentation was done to decide the best way to get accurate results.

II. RELATED WORK

A. Google

As mentioned in the introduction a user can use Google to get similar effects to this search engine. For example if someone is interested in text mining, they may perform a Google search for ”UIUC text mining”. The user may get lucky and get directed to a relevant professors page in one of the top few links. However they may also get a bunch of pages related to text mining research groups and projects at UIUC, instead of the professors.

B. Engineering Expert Search

This search engine can be found at http://engineering.illinois.edu/directory/expert.html. It gives a search engine much like the one I am designed, except it is limited to only engineering professors.

C. Outdated Expert Search

Another example of an existing expert search engine can be found at http://greedy.cs.uiuc.edu/expertsearch/. However any searches result in an Internal Server error, hence the outdated prefix.

III. PROBLEM DEFINITION

The given input to the search engine is fields that the user wants to find professors and faculty are researching. For example, some possible inputs could be ”text mining”, ”machine learning”, or ”distributed systems”.

The expected output is a list of relevant professors and faculty related to the input. The output will also display relevant information to the professor, such as contact information, their office, and anything else that may be necessary.

Before we can create the desired output from our input, we need to have some form of algorithm and index to use. Some web crawling was done to get the raw data needed to perform this algorithm. Then an ElasticSearch index was created after parsing and cleaning up the data.

Essentially we need take the input, and query our ElasticSearch index. Our algorithm will then return a list of relevant documents from the index. We then need to convert the results into the list that represents the output.

IV. METHODS

V. EVALUATION / SAMPLE RESULTS

VI. CONCLUSIONS & FUTURE WORK

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

APPENDIX

All work mentioned in this report was done solely by Alvin Jou.

REFERENCES

[1] List of all UIUC faculty members from http://illinois.edu/ds/facultyListing

[2] Bing Search API to get a list of URLs per faculty member. https://datamarket.azure.com/dataset/bing/searchweb

[3] BeautifulSoup to parse HTML data. http://www.crummy.com/software/BeautifulSoup/

[4] Calaca to create a simple UI for the ElasticSearch index. https://github.com/romansanchez/Calaca