 Computer Science Research Recommendation Service

|  |  |  |  |
| --- | --- | --- | --- |
| Yeounoh Chung | Scott Houde | Johannes Novotny | Erfan Zamanian |
| [yeounoh\_chung@brown.edu](mailto:yeounoh_chung@brown.edu) | [Scott\_houde@brown.edu](mailto:Scott_houde@brown.edu) | [Johannes\_novotny@brown.edu](mailto:Johannes_novotny@brown.edu) | [Erfanz@cs.brown.edu](mailto:Erfanz@cs.brown.edu) |

# Introduction

Every year, a lot of prospective students contact professors before they apply to Ph.D. programs. The goal is not to contact as many professors as possible, but to contact those whom the students want to work with. But, how do you know which professors from which universities are working in the areas that you are interested in? Given the number of professors and the institutions, a brute-force search is not a good idea.

To guide the prospective Ph.D. students in Computer Science in their research advisor search, we propose a research advisor recommender service that we are calling Proffinder. The service will help out the applicants with tailored lists of professors in the areas of interest, specified by a keyword search.

# Data Sets Used

For our project we used three main data sets, 1) computer science faculty data (Huang, n.d.), 2) citation network dataset (Jie Tang, 2008), and 3) Integrated Postsecondary Education Data Set (IPEDS) (National Center of Education Statisics,, n.d.). We will be discussing each of these branches and how we brought them together in the following sections. We also made use of crowdsourcing to obtain the computer science department webpage for each university.

## Computer Science Faculty

In the previous and current iteration of Jeff Huang’s HCI Seminar, students collected profile information of 2,194 CS professors at 55 U.S. universities. Data fields collected per professor included name, rank, year joined the faculty, undergraduate/graduate degree universities subfields and links to additional information (e.g. portrait image, university profile page). While the data went through various verification steps it remains noisy and incomplete in some fields, most importantly the subfield information. In our project this faculty data is used to filter down and our other data sets and in that sense is the main driver for us. Our only alteration to this data set was to add a faculty ID that we generated based on the citation data set and the IPEDS Unit ID that is unique for each institution.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Name* | *Faculty ID* | *IPEDS\_UNITID* | *University* | *Join* | *Rank* | *Subfield* | *BSc University* | *MSc University* | *PhD University* |
| Jeff Huang | 123456 | 987654 | Brown University | 2013 | Assistant | Human-Computer Interaction | University of Illinois at Urbana-Champaign - USA | University of Illinois at Urbana-Champaign - USA | University of Washington - USA |

Table 1: An example record from the faculty dataset

## Citation Network Dataset

This dataset has information about more than 629,814 citations, each associated with abstract, authors, year, venue, and title. The abstract as well as the title can be used to extract research keywords associated with the authors. An extended version of recent citation network dataset includes 4,354,534 citation relationships (as of May 25th, 2014). The dataset contains a lot more examples with missing attribute values (e.g., abstract); however, we use this most recent dataset to report up-to-date research areas of professors.

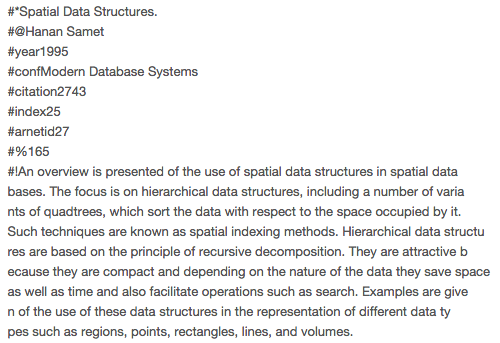


Figure 1 - Citation Network Dataset example

The dataset contains multiple text files for domain, venue, publication, and author information. To generate a research keyword document for each professor, we first draw a relationship among different entities (i.e., different categories of information).

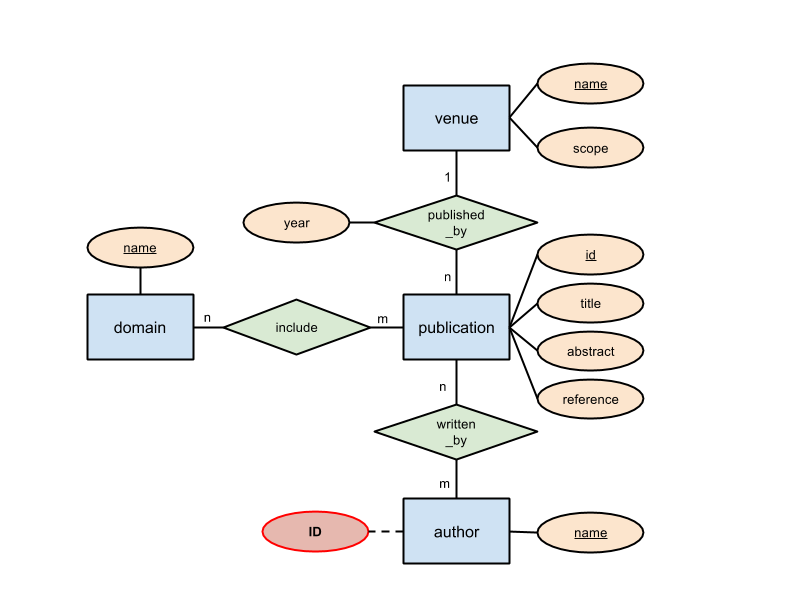


Figure 2. ER-diagram for authors and their research (i.e., domain, publication)

We index all the authors by a unique integer value from 1 to |author|. There will be a single document per author, and all publications related to a given author are concatenated to form a document. A document is represented as a string vector:

d=<id,autname,pubtitle:venname:domname:pubabstract, ...>.

If there are multiple publications by the author, then ‘pubtitle:venname:domname:pubabstract’ for each publication will be joined with the other with a comma in-between.

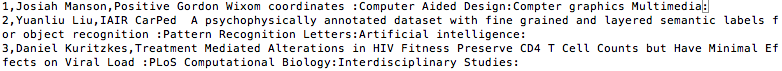


Figure 3. Examples from corpus dataset

Research keywords or fields of their expertise’s can be obtained from various relations and attributes. In specific, we examine domain name, publication titles, abstracts, and publication venue names and scopes for keywords and associate them with authors.

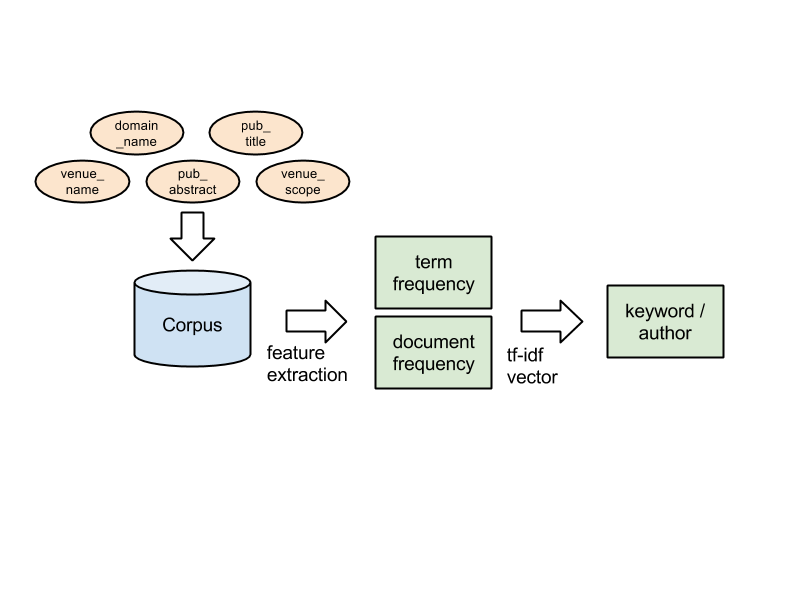


Figure 4. Keyword extraction using tf-idf statistic

A simple keyword extraction procedure is shown in Figure 3. A corpus over all relevant relations and attributes is created; terms (e.g., unigram and bigram) in the corpus are extracted as features as a bag-of-words. Once the bag-of-words model is prepared, we can compute *tf-idf* for each term, and use the *tf-idf* statistic to extract keywords for each author. In brief, term frequency is computed as tf(t,d) =fd(t)maxwdfd(w), inverse document frequency is computed as idf(t,D)=ln(|D||{dD:td}|); we simply multiply those together to get *tf-idf*. In this work, we compute *tf-idf* when we build an inverted index for search term-professors relationship (Section 1.4).

We then created an inverted index for search, which will take in user’s search terms (e.g., research keywords) and output related professor profiles and other information.

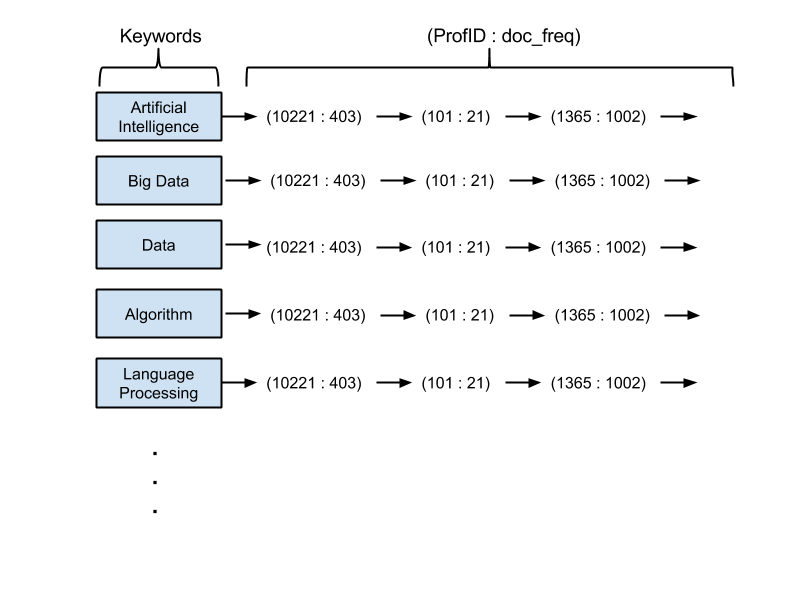


Figure 5. Inverted index for keyword search

The index is based on the research keywords from the corpus generated in the previous step. The relevance of each keyword for a given professor can be measured using tf-idf, and we link each unigram and bigram keywords appear in the corpus to the professor IDs who have published containing the keywords. A term’s *idf* can be computed as an inverse of the length of the linked list; tf is computed as we build the inverted index and annotated next to the professor IDs.

## Integrated Postsecondary Education Data Set (IPEDS)

IPEDS is the Integrated Postsecondary Education Data System. It is a system of interrelated surveys conducted annually by the U.S. Department’s National Center for Education Statistics (NCES). IPEDS gathers information from every college, university, and technical and vocational institution that participates in the federal student financial aid programs. The Higher Education Act of 1965, as amended, requires that institutions that participate in federal student aid programs report data on enrollments, program completions, graduation rates, faculty and staff, finances, institutional prices, and student financial aid. (National Center for Education Statistics, n.d.)

It is a rich source of information about universities in the United States and the information is available for anyone to download from [the IPEDS Data Center](http://nces.ed.gov/ipeds/datacenter/). Each data file comes with a dictionary defining the fields and any other fields, like ALEVEL (IPEDS Award Level) were easily defined after a quick web search. For our project here we used only a small part of the data. The table below shows the three files we used. Our files all came from the Fall 2013 time frame since that is the most recent confirmed data available. In the fields used section I list the items we used in our visualizations, we did not strip out any other data fields in case they became useful later. Each of these files were stripped down to just the rows we wanted using Python scripts. For all of the files that meant reducing by UNITID and the completions file we also eliminated any rows that were not for doctoral degrees in the computer science area, as defined by [CIPC codes](https://nces.ed.gov/ipeds/cipcode/cipdetail.aspx?y=55&cipid=88085)

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | **Survey filename(s)** | **Title** | **Fields Used** |
| Institutional Characteristics | hd2013.csv | Directory information | UNITID, Name, Address City, State, Zip, Lat and Long, Inst web site, Net Price Calculator |
| Enrollments | e2013\_a.csv | Race/ethnicity, gender, attendance status, and level of student: Fall 2013 | UNITID, Level of student (Undergrad, Masters, Doctoral).Enrollments by Gender, Race and Ethnicity |
| Completions | C2013\_a.csv, c2012\_a.csv and c2011\_a.csv | Awards/degrees conferred by program (6-digit CIP code), award level, race/ethnicity, and gender: | UNITID, CIPCode, Award level, counts by the same Gender, Race/Ethnicity and Non-Resident Alien fields as enrollment data set. |

We converted the institutional characteristics file into a GEOJSON object using python scripts for our visualization and also added the computer science department website, generated by crow sourcing and total faculty counts, from the faculty data set.

# Visualization

To provide access our dataset we implemented a web-frontend similar to the Google search engine. Our main webpage provides the basic search field for the research fields users are interested about. Additionally, we provide an overview map visualizing locations (implemented in d3), names and number of computer science faculty for all universities covered by our dataset.  
After searching for a term, we return a results list containing all professors related to the search term ordered by their relevance based on the tf-idf value. Apart from the name, we also list academic rank, current university, subfield, and publication information about each professor. Selecting a professor from this list leads to his/her individual profile page.  
The profile page aggregates detail information from all of our data sources. In the central page we combine personal information of a professor retrieved from Jeff Huang’s dataset (e.g. the year he/she joined the faculty) with a chronological list of publications. General Information about the computer science department, the university and the city in which it is located is shown in the sidebar.

Follow this link (TODO: link ) for a static demo of our web application. It simulates a search for the term “Distributed computing” and a subsequent click on the profile of Maurice Herlihy.

# Challenges

## Citation Challenges

Sourcing complete professors and universities datasets and integrating them all together is a huge challenge that shapes the eventual recommendations in the end. Entity resolution is a well-known problem in data integration; furthermore, given the nature of the information we are dealing with, missing professors or incorrect information cannot be estimated, and what is missing in the base datasets will be missing in the recommendations as well. We were only able to match up 2128 professors out of 2194. Mapping from the user search terms to the entries in the integrated dataset is a challenge in that user can issue any search terms. We have built an inverted index to map a keyword to a list of related professors. The feature space for any search query is enormous, and it impossible to examine all possible matching research keywords for the query. Instead, we only search through the unigrams and bigrams that contain any matching terms in the query.

## IPEDS Data

Despite being considered highly accurate, coming directly from the institutions themselves, transforming the data from CSV to JSON revealed some problems with it. Some rows had different numbers of fields, resulting in bad output due to mis-indexing. Some fields also contained commas when they shouldn’t giving us the same problem. In the end several rows in the final JSON objects were manually altered to be correct, something that would have to be done again if we were to re-run the data to add more institutions.

## Web-Based Visualization

No one on our team has extensive web-based app experience which made setting up the web server and finding the best way to pass information between pages and the like more difficult than we expected. Even knowing better we didn’t allow enough time to really refine the D3 visualizations and make them as interactive as we would have liked.

# Open Questions

There are a number of avenues for future work that would be interesting to explore:

* The essence of our ranking system is based on the publications list. If search term ‘T’ occurs more often in the title of publications of professor A than professor B, then professor A is considered a better fit when the user searches for term ‘T’. While in many cases this could be a good measure of relevance between professors and research areas, oftentimes it could potentially result in misleading rankings. For example, the publications in the area of databases usually do not have the word ‘database’ in their titles. The question here is what should be used instead of/in addition to the publication titles? We believe that paper abstracts could be a good point to start.
* Our recommendation system helps students find professors with desired research interests. However, students are oftentimes interested to dig deeper about the research interests of a particular professor. A word-cloud of most frequent research areas a professor corresponds to could convey a lot of information.
* In a similar vein, when showing the profile of professors, it could be informative for students to find out about other similar professors. For example, when showing Tim Kraska’s profile, the system should show Stan Zdonik, Ugur Cetintemel, Michael Franklin, etc. This would allow users to navigate through the system not only by search terms, but also by research similarity. Rather than keyword-driven, this approach of recommendation is more semantic-driven,
* The research areas of professors are not the only determining factors when it comes to choosing a graduate program. The university itself, its reputation, location, academic/recreational facilities, etc. could influence one’s decision as much. For example, the user might be interested to find faculty members in database area only in the west coast, or in cities with low crime rate, or in universities, which do not have a severe gender ratio problem. Although our recommendation system provides some information about the city in which the university is located (by linking to [www.city-data.com](http://www.city-data.com)), providing these pieces of information in a more structured way (for instance in a way that enables the user to search by them) is a potential future avenue.

# Conclusion

In the end we feel we delivered a very interesting project that has a ton of potential. During our poster presentation many students wanted to come over and try out our system, searching for their particular areas of interest and taking note of the names that appeared. We did not some people felt strongly that our system was “wrong” since the names and relevance did not match up to their expectations. Our system was working accurately, from the information we had available to feed it. If we had more abstracts and more publications available the results would be better, or maybe closer to what people expected anyway.

Our original proposal had the following table of desired deliverables.

|  |  |
| --- | --- |
| **% of Grade** | **Deliverable** |
| 75% | A linked data-set of publication, university and CS faculty profile information with a serviceable but minimal web based user interface with search engine |
| 100% | As above but the user interface will be increased by making better use of visualizations and interactive searching, potential inclusion of a recommendation engine on top of the search engine. |
| 125% | In addition to the base data set and a highly usable web interface additional data sources will be pulled in such as faculty social media like Twitter or LinkedIn.  A recommendation engine will be able to make suggestions of faculty like other faculty based on publication abstract keyword comparison. |

We believe that we have firmly met the first two goals. We cleaned the data sets, transformed some data into a more useable format and made a visible web based search engine. However we feel that we feel short of the final goal. In the end we have many of the pieces we needed in place but we just ran out of time to assemble them. The visualization was not to where we had hoped it would be, and although we were shorting our search results by relevance we were unable to implement a feature like “find other professors like this one”. Given some more time we are quite confident we could have delivered something both a joy to use and very useful.