 Integrating Data Sets to Provide Search Tools for Computer Science Doctoral Program applicants

By

|  |  |  |  |
| --- | --- | --- | --- |
| 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) |

# Abstract

# 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.

## 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.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *Name* | *University* | *Join* | *Rank* | *Subfield* | *BSc University* | *MSc University* | *PhD University* |
| Jeff Huang | 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 |

An example record from this 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.

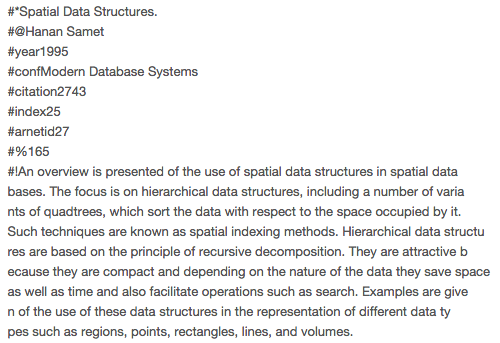


Figure 1. Citation Network Dataset example

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).

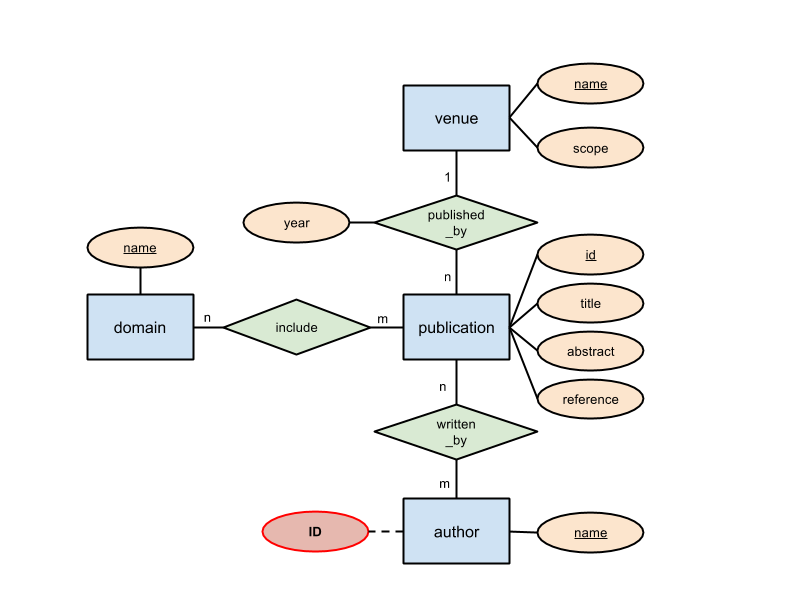


Figure 2. ER-diagram for authors and their research (i.e., domain, publication)

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.

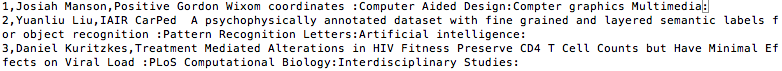


Figure 3. Examples from corpus dataset

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.

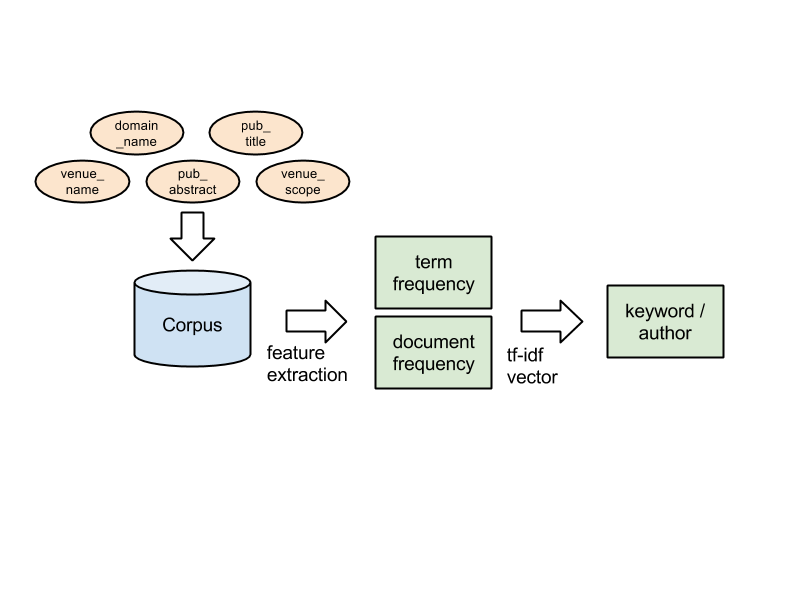


Figure 4. Keyword extraction using tf-idf statistic

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).

## 1.4 Inverted Index

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

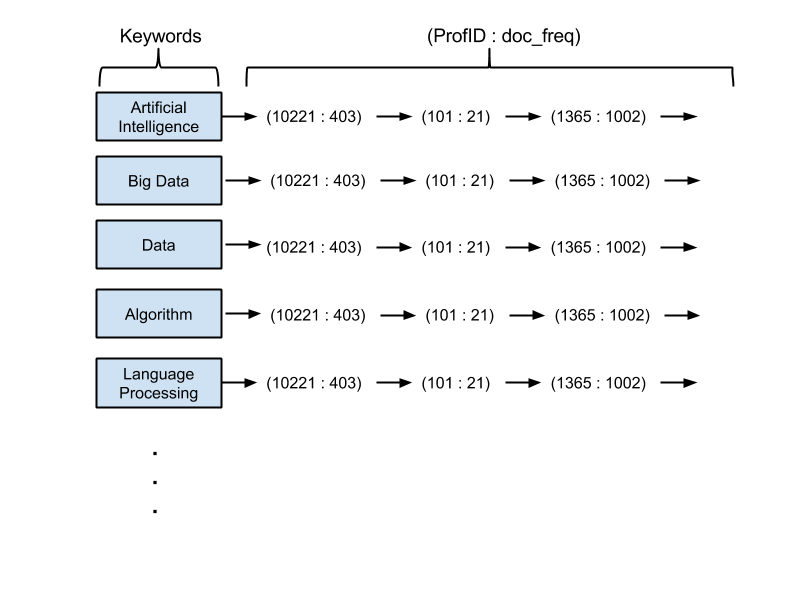


Figure 5. Inverted index for keyword search

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)

# Visualization

# Open Questions

# Conclusion

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. EXPLAIN HOW