INFSCI 2930 - INDEPENDENT STUDY

# Building a Knowledge Graph for Recommending Experts

**Behnam Rahdari**
(behnam.rahdari@pitt.edu)

*Summer 2019*

# Table of Contents

## Abstract

Recommending experts is an important challenge in many contexts. In this paper, we present a method to build a knowledge graph by integrating data from Google Scholar and Wikipedia to help students find a research advisor or thesis committee member. This knowledge graph is used to power the exploratory search interface to recommend similar keywords and relevant scholars to the students with a limited level of knowledge and familiarity with the subject of research.


## Introduction

Recommending experts is an important challenge in many contexts. The nature of this challenge is to find a knowledgeable person with an advance expertise in one or more target topics among a large number of potential candidates. A well-explored example is finding an expert for a specific project within a large company or finding a doctor with advance knowledge of a specific disease in a large city. While in these two contexts large companies and hospitals use knowledge management techniques to catalogue key areas of expertise and use it to represent information about each expert, finding experts in other contexts could be more challenging.

The context that we target out in this paper is finding a research advisor. Every year many undergraduate, master-level and PhD students are facing a difficult of finding a research advisor. While large universities have many highly knowledgeable faculties, finding one with the expertise that matches student interests, requirements, and preparation is a hard problem. Whether this search is for finding an advisor for a summer research project, a faculty sponsor for an independent study, or a committee member for a PhD thesis, online sources frequently fail the students and they resort to the word of mouth within a limited circle of instructors, classmates, and university staff. One problem is a wide variety of sources where advisor information could be found online (department directories, publication sites, funding agency pages, personal home pages, etc.) Each of these sources covers only some aspects of faculty expertise and frequently represent only a subset of available advisors. Another problem is the lack of "expertise catalog". A university usually offers a catalog of courses and majors, but not a fine-grained catalog of expertise areas covered by faculty. As a result, students frequently can’t even properly name their target area of interest or formulate a Web search query when looking for advisors.

The focus of our project is to offer a single-access-point exploratory search system, which allows students to discover their target areas of interest and find relevant advisors within these areas. In its core, the platform uses a knowledge expertise graph, which represents multiple connections between research topics and prospective research advisors within a large university or a large research field. We build this graph by processing several knowledge sources about faculty and their research interests. This paper briefly reviews the kind of knowledge graph we built, the process of its building by information extraction, and the information exploration system powered by this knowledge.

## Background

The attempts to build "a map of science" representing most important areas of research expertise and their connections with experts have been done in the past, however, the lack of proper information sources made it hard to produce maps that are suitable for finding advisors. A good example are where academic journals are used as proxies of expertise areas and a map of science is built by clustering journals by co-publication links. While this map is useful as a "big picture" of science, its use for advisor finding is problematic since it represents expertise on a very coarse-grain level and misses many prospective advisors who are not frequent journal authors. However, the emergence of modern sites powered by a combination of advanced information processing and collective wisdom makes the task of building a fine-grain knowledge network of experts and expertise areas feasible. In our work we rely most extensively to two of these sites - Google Scholar and Wikipedia.

*Google Scholar* has been long recognized among the best freely accessible academic information sources in term of coverage and accessibility. It has been compared positively with number of similar citation services namely *Web of Science*, *PubMed* and *Scopus*. Yet, although *Google Scholar* contains nearly 160 million documents covering a large portion of published documents, the lack of semantic connections between concepts and keywords within these documents makes it difficult to use the system for finding advisors, especially by less experienced students.

*Wikipedia* is commonly used by researchers to compute the semantic relatedness within documents, extract Open Information and mine meaning using relations, facts and description to extracts and makes use of the concepts.

## Building the Knowledge Graph

To support students in finding advisors we created a knowledge graph using data from Google Scholar and enriching it semantically using Wikipedia. In turn, this graph was used to power an interactive exploratory recommendation interface, which make the task of advisor-finding easier, especially for students with a limited level of knowledge and familiarity with the subject of research. To support several advisor-finding scenarios, we build several versions of the graph. The graph presented in this paper is focused on task of finding a top expert in a specific topic of interest within some broad field of research (such as Artificial Intelligence) across many universities. This is a typical task for a student selecting a PhD program to join or for a senior PhD student looking for an external thesis committee member.

## Data Sources

| | Artificial Intelligent | | Computer Architecture | |
|---|---|---|---|---|
| | all | unique | all | unique |
| Self-defined Keywords | 1916 | 628 | 1671 | 493 |
| Publication Concepts | 5946 | 1985 | 5889 | 1650 |
| Relevant Keywords | 37339 | 24712 | 30677 | 21355 |
| Wikipedia Categories | 3488 | 857 | 2496 | 1775 |
| Co-Author Relationship | 5727 | 4771 | 5096 | 3287 |

**Table 1**: *Data Statistics: number of items for each field*

### Google Scholar

We exploited the information of 1000 active scholars in two popular fields of computer science; *Artificial Intelligent* and *Computer Architecture* (focusing on top 500 scholars per field). For each individual, we extracted the following information (see Table 1):

- **Name**: Full name of the scholar.
- **Affiliation**: The university or research institution the scholar affiliated with.
- **Verified Email Domain**: It used to check the validity of the scholar profile.
- **Self-Defined Keywords**: A list of up to five keywords defined by scholars to describe their research interests.
- **Citations**: The total number of citations received by all the scholar's publications.
- **h-index**: The h-index measures the citation impact and productivity of a scholar's publications. We use this measure alongside with other quantitative scores to re-order the results of the recommendations.
- **i10-Index**: i10-Index describes as the total number of scholar's publications with 10 citations and more. This score, which is only used by Google Scholar, also used to re-rank the results of the recommendations.
- **Recent publications (20)**: We used 20 most recent publications to generate additional keywords representing current interests of each scholar. The keywords were extracted from the titles of recent publications as follows: After removing stop-words, we generated all the possible keyword candidates as unigrams, bi-grams, and tri-grams. Next, we only kept the keywords that have an entry in Wikipedia (see keyword verification below).
- **Top Co-Authors (10)**: For each scholar, we extracted a list of top 10 co-authors from their Google Scholar profile.

### Wikipedia

We used Wikipedia to add a semantic layer to profiles extracted from Google Scholar. Throughout this process, we also obtained some useful information that led to a stronger connection between keywords and enables us to add weight to each scholar-keyword relation. Wikipedia API has been used for the following purposes:

- **Keyword verification**: As mentioned before, we collected two sets of keywords for each scholar: self-defined keywords and keywords extracted from recent publications. Wikipedia API has been used to verify the validity of these keywords by using fuzzy match technique to find the Wikipedia entry describing the keyword. We removed all keywords that did not match with any article in Wikipedia. While Wikipedia might miss articles for some less popular research topics, we need to have all topic keywords explainable for the student audience and a match to a Wikipedia article was the best way to assure it. For all remaining keywords, we calculated association weight between a keyword and a scholar as cosine similarity between the full-text Wikipedia entry of each keyword and concatenated text from scholar's recent publications.
- **Entry Summary**: To offer student users a short description of each topic keyword, we collected page summary for all keywords using Wikipedia API.
- **Top relevant keywords (10)**: Most (if not all) Wikipedia pages have multiple links to similar or related articles. We collected the top 10 links based on the number of their occurrences in each page. We employ these links to create a highly connected network of keywords.
- **Entry Categories**: Wikipedia uses categories to group similar articles. We extracted all categories associated with a page and used a full Wikipedia category hierarchy schema to find relationships between categories in our data set.

## Graph Representation

We used Neo4j graph database to represent information about all scholars and keywords. The overall schema of the knowledge graph is represented in Figure 1. As is shown, there are three distinct node types in the graph:

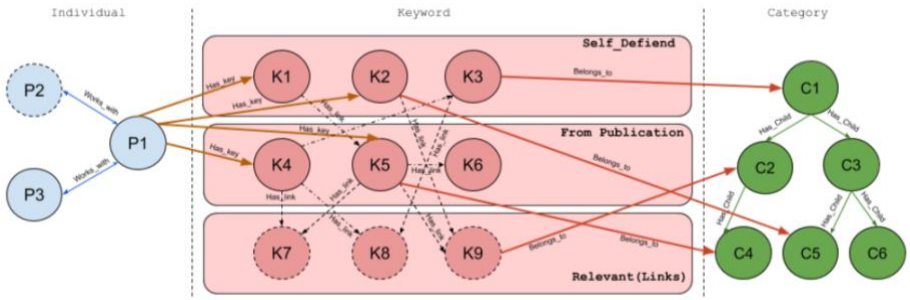


**Figure 1**: *High level Knowledge Graph schema: from left to right, "individual nodes" (blue) store scholar's demographic information, "keyword nodes" (red) keep detailed information about topic keywords and "category nodes"(green) convey the hierarchical association between categories and semantic relationship between keywords*

- **Individual**: This node type conveys demographic information about scholars including full name, affiliation, verified email domain, and URLs of personal homepage and Wikipedia page (if exist). *Individual* nodes are connected via *"works_with"* links which represent the co-authorship relations between scholars. An *individual* node also connects to several *keyword* nodes that represent the scholar's research interests and expertise. The *individual* nodes with a dashed border represent scholars added via *co-authorship extraction* who are

not among the top 500 extracted scholars. These nodes are not considered in the final recommendations and only used to indicate the connections of the top scholars.

- **Keyword**: There are three types of *keyword* nodes. Self-Defined keywords, keywords extracted from recent scholar's publications, and relevant keywords that are represent the connection between two other types (shown by a dashed border) and will not appear in the recommendations. The relationship between keywords represented by *"has_link"* arc is established if the target node has been mentioned in the source node's entry page. *Keyword* nodes are connected to *individual* nodes via *"has_key"* and to *category* nodes via *"belongs_to"* arcs.
- **Category**: We employed a full hierarchical schema of Wikipedia categories to represent the inter-connectivity between categories in our data set. These relations are presented as *"has_child"* arc in Figure 1 The *category* nodes are used to find the semantic relationship between keywords.

# Using the Knowledge Graph

## Interface Design

The knowledge graph is used to power the exploratory search interface for finding advisors. the interface consists of four main sections.

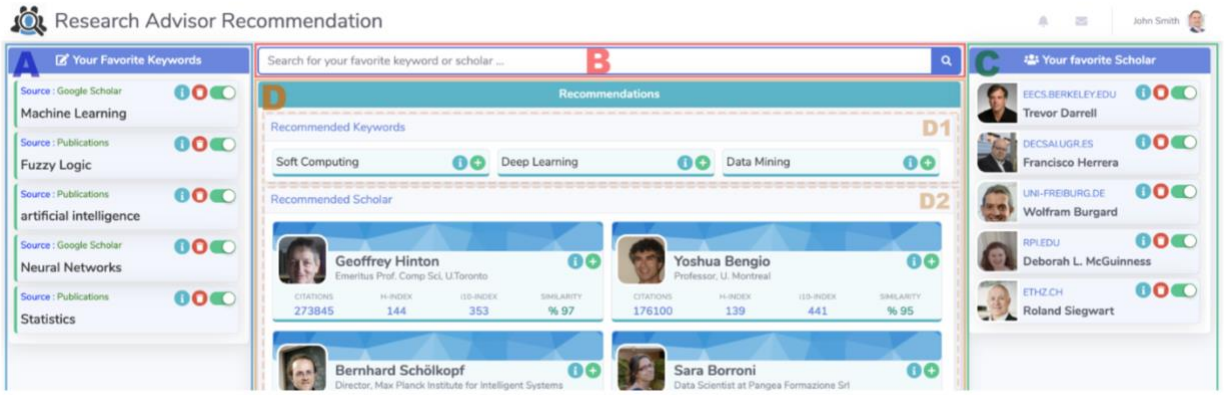


**Figure 2**: *Interface Design of Research Advisor Recommendation*

## Instant search box

User can use the search box to search for topic keyword or scholars of interest (Figure 2: B). When the user starts typing, a list of matching keywords and scholars will appear. When an item is selected from the list, it will automatically be added to proper location on left or right side of the interface. at the same time, an updated list of recommendations will be presented to the user.

## Favorite keywords

This section (Figure 2: A) shows user's "favorite" keywords. User add keywords to this list using the instant search box or by clicking on the *plus button* next to each recommended keyword. User can interact with three buttons on the right side of each keywords to (1) see more information about that keyword (including Wikipedia summary, similar keywords and other scholars with this

research interest), (2) remove the keyword from the favorite list and (3) enable/disable the effect of this keyword in the list of recommendations.

### Favorite scholars

Similar to *favorite keywords*, the user can assemble a list of favorite scholars (Figure 2: C). A new favorite scholar could be added to the list from the instant search results or a list of recommended scholars by clicking on *plus button* next to a recommended scholar. The three buttons on the right side of each favorite scholars could be used to obtain more details about the scholar (affiliation, full list of research interests, and similar scholars), remove the scholar from the list, and enable/disable the effect of this scholar in the final recommendation. Together with the *favorite keywords* list explained above, the list of favorite scholars forms a user *profile of interests*, which the user gradually assembles while exploring possible areas of interests and scholars. In turn, the profile of interests is used to generate further recommendations as explained below.

### Recommendations

This section (Figure 2:D) consist of two subsections. *Recommended keywords* (Figure 2: D1) shows the list of three most relevant additional keywords, which are suggested given already selected (and enabled) user's favorite keywords and scholars. User can see more information about the keyword (similar to favorite keyword section) and also add this recommended keyword to their favorite list using two circular buttons on the right side of each keyword. *Recommended scholars* (Figure 2: D2) shows a list of recommended scholars, which are most relevant to the active (enabled) favorite topic keywords and most similar to the active favorite scholars. For each recommended scholar the list shows basic personal and academical information. User can also see the similarity between the recommended scholars and their profile of interests represented by favorite keywords and scholars.

## Recommendation method

We generate the recommendations using "Cypher Query Language" in Neo4j. In the following we explain how we generate recommendations for keywords and scholars.

### Keyword Recommendation

In order to recommend similar keywords, we use user's favorite keywords and scholars. Each keyword is connected to other keywords in two ways: 1- Via the similar research interest between scholars and 2- Via similar relevant keywords and categories. We consider both of these relations to find similar keywords. In the final list, we sorted the keywords based on the number of occurrences then we chose top three keywords to be presented to the user.

### Scholar Recommendation

Similar to keyword recommendation, we use both favorite keywords and scholars. There are three criteria for scholar recommendation: weighted scholar's research interests, co-authorship relationship between scholars, and connection between scholar interest through relevant keywords and categories. After generating the list of candidate scholars, we sort it based on the similarity score (calculated based on weighted similarity score for each of three criteria) and present the top ten results to the user.

## Discussion and Future Work

We presented a method to build a knowledge graph by integrating data from Google Scholar and Wikipedia to help students with limited knowledge about the subject find a research advisor or thesis committee member. Although Google scholar covers a variety of publications and patents, additional sources of information (scholar's active research projects, funding information, etc.) could make the knowledge graph more connected and provide the users with additional critical information when it comes to finding an advisor. We also plan to refine our keyword extraction technique. More sophisticated methods of extraction using natural language processing and machine learning could potentially improve the semantic relation between concepts and provide users with a more realistic set of research interests for scholars. We also designing a series of controlled user study and field studies to evaluate the usability and value of the exploratory search interface. We hope, that these user studies will provide valuable insights for improving the knowledge graph and the interface.