

Building a Knowledge Graph for Recommending Experts

Behnam Rahdari^[0000-0001-6514-912X] and Peter Brusilovsky^[0000-0002-1902-1464]

University of Pittsburgh, Pittsburgh PA 15260, USA
{ber58,peterb}@pitt.edu

Abstract. Identifying 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.

Keywords: Data Integration, Knowledge Graph, Recommender Systems

1 INTRODUCTION

Identifying 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 of this task 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 in this paper involves students finding a research advisor. Each year, undergraduates, master-level and doctorate students face the difficult challenge of finding a research advisor. While large universities have many highly knowledgeable faculty, finding one with the expertise that matches the student's interests, requirements, and preparation is a challenging task. Whether the task is finding an advisor for a summer research project, a faculty sponsor for an independent study, or a committee member for a doctoral

DI2KG 2019, August 5, 2019, Anchorage, Alaska. Copyright held by the author(s). Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0)

thesis, online sources frequently fail the students, and they resort to 'word of mouth' within a limited circle of instructors, classmates, and university staff. One problem in using online sources is the wide variety of sources with relevant information that can exist (e.g., department directories, publication sites, funding agency pages, personal home pages, etc.) Each of these sources covers only some aspects of the faculty member's expertise and frequently represent only a subset of available advisors. Despite these different sources, there is typically a lack of "expertise catalogs". 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 cannot 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 built this graph by processing several knowledge sources about faculty and their research interests. This paper briefly reviews the type of knowledge graph we built, the process of extracting information for its development, and the information exploration system powered by this knowledge.

2 BACKGROUND

In the past, there have been attempts to build "a map of science" representing most important areas of research expertise and their connections with experts; however, the lack of proper information sources makes it hard to produce maps that are suitable for finding advisors. Examples of this attempt to build a map of science is presented in [12] and [2], 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 in the context of finding research advisors is problematic since it represents expertise on a very coarse-grain level and does not capture 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 on two of these sites - *Google Scholar* and *Wikipedia*.

Google Scholar has been long recognized as one of the best freely accessible academic information sources in terms of coverage and accessibility [11]. It has been compared positively with a number of similar citation services namely *Web of Science* [6], *PubMed*, and *Scopus* [7, 5]. Yet, although *Google Scholar* contains nearly 160 million documents [3] 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 of concepts between and within documents [9, 8, 1], extract Open Information [4], and mine meaning using relations, facts, and descriptions to extract and use concepts [10].

3 BUILDING THE KNOWLEDGE GRAPH

To support students in finding advisors, we created a knowledge graph using data from Google Scholar and enriched it semantically using Wikipedia. In turn, this graph was used to power an interactive exploratory recommendation interface, which makes 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 built several versions of the graph. The graph presented in this paper is focused on the 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 doctoral program to join or for a senior doctoral student looking for an external thesis committee member.

3.1 Data Sources

Google Scholar We utilized the information of 1000 active scholars in two popular fields of computer science: *Artificial Intelligent* and *Computer Architecture* (focusing on the top 500 scholars in each 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 is affiliated with.
- **Verified Email Domain:** 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 of 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 other quantitative scores to re-order the results of the recommendations.
- **i10-Index:** i10-Index describes the total number of the scholar’s publications with 10 citations or more. This score, which is only used by Google Scholar, was also used to re-rank the results of the recommendations.
- **Recent publications (20):** We used the 20 most recent publications to generate additional keywords representing the current interests of each scholar. The keywords were extracted from the titles of recent publications as follows: After removing stop-words, we generated all of the possible keyword candidates as uni-grams, 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 the top 10 co-authors from their Google Scholar profile.

	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 item for each field

Wikipedia We used Wikipedia to add a semantic layer to profiles extracted from Google Scholar. Throughout this process, we also obtained useful information that led to a stronger connection between keywords and enables us to add weight to each scholar-keyword relation. The 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. The Wikipedia API has been used to verify the validity of these keywords by using fuzzy match techniques 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 explained for the student audience and a match to a Wikipedia article was the best way to assure it. For all remaining keywords, we calculated the association weight between a keyword and a scholar as cosine similarity between the full-text Wikipedia entry of each keyword and concatenated text from the scholar’s recent publications.
- **Entry Summary**: To offer student users a short description of each topic keyword, we collected page summaries for all keywords using the 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.

3.2 Graph Representation

We used the 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:

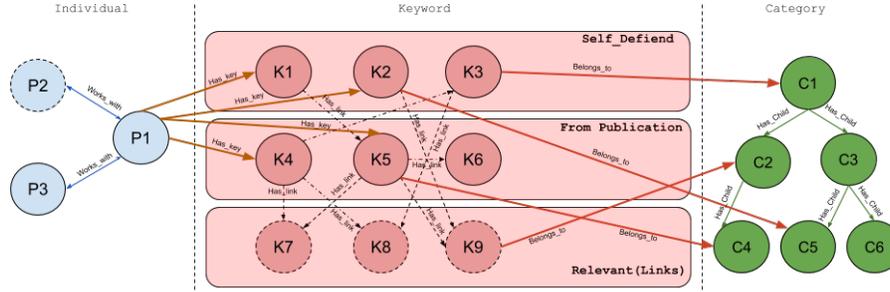


Fig. 1. High level Knowledge Graph schema: from left to right, "individual nodes" (blue) store the scholar's demographic information, "keyword nodes" (red) keep detailed information about topic keywords, and "category nodes" (green) convey the hierarchical association between categories and the semantic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

- **Individual:** This node type conveys demographic information about scholars including full name, affiliation, verified email domain, and URLs of personal homepages and Wikipedia pages (if they 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 publications, and relevant keywords that 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 the "has_child" arc in Figure 1. The *category* nodes are used to find the semantic relationship between keywords.

<https://en.wikipedia.org/wiki/Neo4j>

4 USING THE KNOWLEDGE GRAPH

4.1 Interface Design

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

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

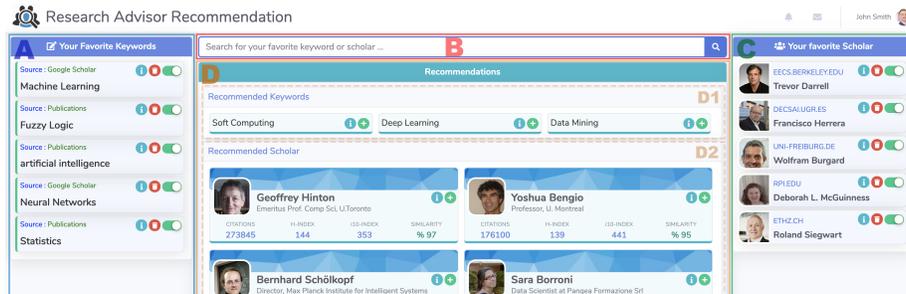


Fig. 2. Interface Design of Research Advisor Recommendation

Favorite keywords This section (Figure 2:A) shows users’ “favorite” keywords. Users add keywords to this list using the instant search box or by clicking on the *plus button* next to each recommended keyword. Users can interact with three buttons on the right side of each keywords to (1) see more information about that keyword (including its 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 on the list of recommendations.

Favorite scholars Similar to *favorite keywords*, users can assemble a list of favorite scholars (Figure 2:C). A new favorite scholar can be added to the list from the instant search results or a list or recommended scholars by clicking on the *plus button* next to a recommended scholar. The three buttons on the right side of each favorite scholar can 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 form the users' *profiles of interests*, which the users gradually assemble while exploring possible areas of interests and scholars. In turn, the profiles of interests are used to generate further recommendations as explained below.

Recommendations This section (Figure 2:D) consists of two subsections. *Recommended keywords* (Figure 2:D1) shows the list of the three most relevant additional keywords, which are suggested given already selected (and enabled) favorite keywords and scholars. Users can see more information about the keyword (similar to the favorite keywords section) and also add these recommended keywords to their favorite lists 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 academic information. Users can also see the similarity between the recommended scholars and their profiles of interests represented by favorite keywords and scholars.

4.2 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 Recommendations In order to recommend similar keywords, we use the 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 the top three keywords to be presented to the user.

Scholar Recommendations Similar to keyword recommendations, we use both favorite keywords and scholars. There are three criteria for scholar recommendations: the scholar's weighted research interests, co-authorship relationship between scholars, and connection between the scholar's interests 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 the three criteria) and present the top ten results to the user.

5 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 a

subject find a research advisor or thesis committee member. Although Google scholar covers a variety of publications and patents, additional sources of information (e.g., the 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 plan to refine our keyword extraction techniques. More sophisticated methods of extraction using natural language processing and machine learning could potentially improve the semantic relations between concepts and provide users with a more realistic set of research interests for scholars. We have also designed a series of controlled user studies 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.

References

1. Behnam Rahdari and Peter Brusilovsky. 2019. User-controlled hybrid recommendation for academic papers. In *Proceedings of the 24th International Conference on Intelligent User Interfaces: Companion*. ACM, 99–100.
2. Colin Murray, Weimao Ke, and Katy Börner. 2006. Mapping Scientific Disciplines and Author Expertise Based on Personal Bibliography Files. In *Tenth International Conference on Information Visualisation (IV’06)*. 258–263.
3. Enrique Orduña-Malea, Juan Manuel Ayllón, Alberto Martín-Martín, and Emilio Delgado López-Cózar. 2014. About the size of Google Scholar: playing the numbers.
4. Fei Wu and Daniel S. Weld. 2010. Open Information Extraction Using Wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL ’10)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 118–127.
5. John Mingers and Martin Meyer. 2017. Normalizing Google Scholar data for use in research evaluation. *Scientometrics* 112, 2 (2017), 1111–1121.
6. Joost CF De Winter, Amir A. Zadpoor, and Dimitra Dodou. 2014. The expansion of Google Scholar versus Web of Science: a longitudinal study. *Scientometrics* 98, 2 (2014), 1547–1565.
7. Matthew E. Falagas, Eleni I. Pitsouni, George A. Malietzis, and Georgios Pappas. 2008. Comparison of PubMed, Scopus, Web of Science, and Google Scholar: Strengths and weaknesses. *The FASEB Journal* 22, 2 (2008), 338–342.
8. Max Völkel, Markus Kröttsch, Denny Vrandečić, Heiko Haller, and Rudi Studer. 2006. Semantic Wikipedia. In *Proceedings of the 15th international conference on World Wide Web*. 585–594.
9. Michael Strube and Simone Paolo Ponzetto. 2006. WikiRelate! Computing semantic relatedness using Wikipedia. In *AAAI*, Vol. 6. 1419–1424.
10. Olena Medelyan, David Milne, Catherine Legg, and Ian H. Witten. 2009. Mining meaning from Wikipedia. *International Journal of Human-Computer Studies* 67, 9(2009), 716–754.
11. Philipp Mayr and Anne-Kathrin Walter. 2007. An exploratory study of Google Scholar. *Online Information Review* 31, 6 (2007), 814–830
12. Richard M. Shiffrin and Katy Börner. 2004. Mapping knowledge domains. *Proceedings of the National Academy of Sciences* 101, suppl 1 (2004), 5183–5185.