Taxonomy-based Query-dependent Schemes for Profile Similarity Measurement

Suppawong Tuarob[†], Prasenjit Mitra[†‡] and C. Lee Giles[†‡]
[†] Computer Science and Engineering, [‡] Information Sciences and Technology
The Pennsylvania State University
University Park, PA 16802
suppawong@psu.edu, {pmitra, giles}@ist.psu.edu

ABSTRACT

Semantic search techniques have increasingly gained attention in information retrieval literature. Authors are great sources of semantic interpretation for documents, especially in scholarly domains where articles mostly reflect the research interests of the authors. Being able to interpret semantic meanings of documents from their authors would give rise to many interesting applications, especially in academic digital library literature. In this paper, we present taxonomy-based query-dependent schemes for computing author profile similarity. Our schemes have the capability to capture partial similarities, as opposed to traditional topic overlap based approaches. We generalize our schemes so that they can be easily adopted to other application domains. We acquire resources from multiple places such as Wikipedia, Citeseer^X, ArnetMiner, and WikipediaMiner as part of our work. We provide encouraging anecdotal results along with suggestions on potential applications of the proposed schemes.

Keywords

Search, Entity Similarity, Profile Similarity

1. INTRODUCTION

In traditional document retrieval processes, textual similarity scores are calculated between the query and documents, then the search engine ranks and returns the results based on such scores. Such method works quite well for most documents that can be represented using bags of words. However, some documents have unique properties that allow the search process to infer further semantic meanings beyond just textual similarity. Search engines for web pages harvest the link connections to infer the importance of web pages using popular algorithm such as PageRank[11] and HITS[10]. People searches in social networks such as Facebook or Google+ utilize the social connections among users to suggest people. Academic search engines such as Citeseer^X[1] also harvest citation networks and co-authorship networks to rank documents.

Scholarly authors are a very rich source for mining semantics for their documents. Obviously authors tend to write articles inspired by their interests and backgrounds. Hence, being able to identify authors' interests and backgrounds would be beneficial to applications in academic document retrieval literature. Recent works on mining authors' research interests have been explored by Tang et al.[13] using a topic modeling approach. Their approach has been implemented in ArnetMiner.org[2] to mine researchers' research interests. Being able to determine the similarity between two authors could be beneficial in determining semantic similarity of the articles written by them. The capability to compute profile similarities could also give rise to many interesting applications such as expertise searching, author ranking, and document recommendation. In different domains such as social networks, profile similarity plays a big role in people recommendation (friend finding), post recommendation, and people ranking.

1.1 Problem Definition

Even though we are interested in scholarly domains, we generalize the problem so that it can be used as a framework in other domains. We define our problem as following. Given a topic library T, a user profile P_U of user U is described by a set of weighted topics

$$P_U = \{<t_{u1}, w_{u1}>, ..., <t_{un}, w_{un}>\}$$

where $\{t_{u1}, ..., t_{un}\} \subseteq T$ and $\{w_{u1}, ..., w_{un}\}$ are real numbers between 0 and 1. A query Q is a set of weighted topics.

$$Q = \{<t_{q1}, w_{q1}>, ..., <t_{qk}, w_{qk}>\}$$

where $\{t_{q1}, ..., t_{qk}\} \subseteq T$ and $\{w_{q1}, ..., w_{qk}\}$ are real numbers between 0 and 1. Our goal here is to compute the similarity score between two user profiles P_A and P_B given a query Q. Formally, we aim to compute $ProfileSimilarity(Q, P_A, P_B)$, a function that returns a real number between 0 and 1, representing the level of profile similarity.

A naive way to compute such profile similarity can be achieved by counting the number of common topics between the two profiles. Such approach seems promising and intuitive; however, it lacks the power to capture partial similarities. Consider the following example. Author A identifies *Machine translation* as her interest. Author B identifies *Random Forest* as her interest. Note that *Machine translation* and *Random Forest* are both *Machine Learning* algorithms. With topic overlapping based methods, both authors A and B would be reported not having anything in common. However, in the real world, these two people may have common interest in machine learning literature. Thus, being able to infer partial similarity between two profiles would improve the accuracy in computing similarity measure between two authors.

[1] http://citeseer.ist.psu.edu

[2] http://arnetminer.org

Copyright is held by the author/owner(s).
JIWES, SIGIR 2012 Workshop, Portland, Oregon, USA

In this work, we propose 10 schemes for computing similarities between two profiles. We divide the schemes into 3 families: topic overlap based, summation based, and maximization based. The topic overlap based schemes only measure the topic overlapness between two given profiles. The summation based schemes sum over the similarity of each pair of topics between two profiles, and compute the average. The maximization based schemes pick the pair of topics between the two users that maximizes the similarity score. The formal definitions of all the 10 schemes are given in Section 4.2. These schemes rely on the existence of a taxonomy of topics, which we will describe in more detail in Section 3.

1.2 Our Contributions

This work has the following key contributions:

1. We propose 10 variants of query-dependent taxonomy-based schemes, divided into 3 families, for computing the similarity measure between two user profiles given a query. Each user profile is described with a set of weighted topics as defined in Section 1.1. Our schemes rely on the taxonomy of topics to infer partial similarity between two given topics.

2. We harvest and combine resources from Wikipedia[3], Citeseer^X, Arnetminer.org, and WikipediaMiner[4] in our research. We retrieve and extract the list of topics and their hierarchy relationship from Wikipedia. We obtain the database of authors along with their publications from Citeseer^X repository. We obtain each author's research interest from ArnetMiner. Finally, we use the tool provided by WikipediaMiner to extract topics from research interests and build a profile for each author.

3. We provide anecdotal results from our experiment among 34 authors from 9 computer science disciplines, using the paper "TextTiling: segmenting text into multi-paragraph subtopic passages"[7] as the query. The results, though not from aggressive experiments, are encouraging and show great promises on a good foundation for future work on the problem.

4. We make suggestions on how to adopt our proposed schemes to useful applications in real world.

2. RELATED WORKS

The literature on mining similarity among entities is extensive, hence we only describe the works closely related to ours. The existing schemes for computing the similarity among entities can be divided into two groups: graph based and content based.

Graph based approaches utilize the relationship between users or the network structures to infer the similarity. Jaccard similarity[12], for instance, is computed based on the tuition that the level of similarity between two nodes correlates with the number of common friends. SimRank[8] is a global similarity measure based on the intuition that two nodes are similar if they are related to similar nodes. Chen et al.[3] propose a generalized node similarity measure, the Relation Strength Similarity (RSS), which is an asymmetric scheme and can be used in weighted networks. The RSS scheme was used in [4] to compute similarities between researchers for collaboration recommendation. The scheme is based on the number of articles co-authored by two given authors. Gollapalli et al.[5] apply PageRank algorithm on the co-authorship network for ranking authors.

Content based approaches rely on the assumption that entities have documents attached or linked to them. Gollapalli et al.[6] explore different content similarity measures namely Okapi BM25, KL Divergence, LDA-based probabilistic modeling, and Trace-based Similarity, to compute similarity between two given authors. Tang et al.[13] propose the Author-Conference-Topic (ACT) where each author is associated with a multinomial distribution of topics. The model is implemented in ArnetMiner as part of the expertise search service.

3. RESOURCES

Harvesting resources from different places and combining them together is a major contribution of our work. This section describes how we obtain the taxonomy of topics which we use as our topic library from Wikipedia, the authors database from Citeseer^X, and the research interests which we use to build a profile for each author from ArnetMiner. We also describe how we extract topics from both the query and authors' research interests using the WikipediaMiner annotation tool.

3.1 Taxonomy of Topics from Wikipedia

Our system relies on the hierarchy relationship between topics to compute partial similarity between two topics. In this work, we extract the taxonomy of topics from Wikipedia, an online encyclopedia supported by the non-profit Wikimedia Foundation. Wikipedia provides categories of the articles which are organized as a directed acyclic graph (DAG). We extracted 758,336 categories along with their hierarchy relationship to build our topic taxonomy.

Obtaining the raw data. We extract the taxonomy of topics from Wikipedia *Page* and *Categorylinks* tables. The *Page* table contains the meta information of all the Wikipedia pages (or articles) such as page IDs (*page_id*), page titles (*page_title*), page types (*page_namespace*), etc. Refer to [1] for more information about the *Page* table schema. The *Categorylinks*[2] table contains the information about what categories each Wikipedia page belongs to. We filtered only "category pages" (i.e. *page_namespace* = 14), and extract the category relationship between such pages.

Cleaning the taxonomy. Even though Wikipedia claims that their categorization is a DAG, the raw taxonomy of topics that we extract still contain cycles. We found 1,757 cycles in the original taxonomy. We hence remove such loops by performing a depth first traversal from the root node and eliminating the links that point back to the nodes already visited. After the cleaning process, we obtain a taxonomy of 758,336 topics. Figure 1 illustrates an example taxonomy taken from a subset of the full taxonomy.

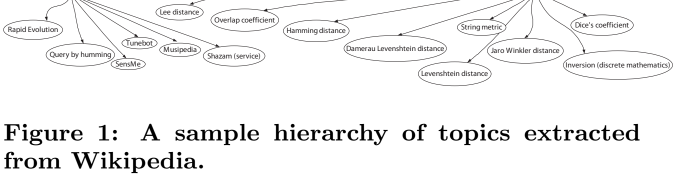
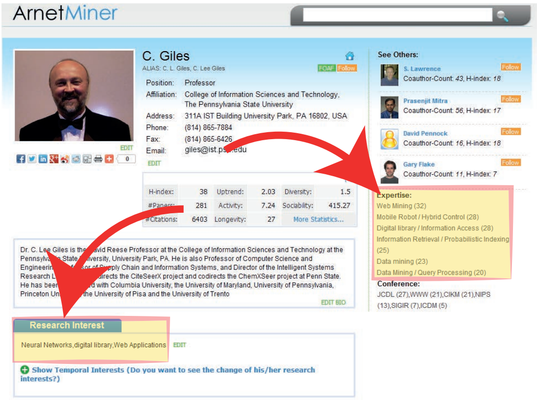


Figure 1: A sample hierarchy of topics extracted from Wikipedia.

[3] http://www.wikipedia.org/

[4] http://wikipedia-miner.cms.waikato.ac.nz/



ArnetMiner

C. Giles

ALIAS: C. L. Giles, C. Lee Giles

FOAF Follow

Position: Professor

Affiliation: College of Information Sciences and Technology, The Pennsylvania State University

Address: 311A IST Building University Park, PA 16802, USA

Phone: (814) 865-7884

Fax: (814) 865-6426

Email: giles@ist.ps .edu

EDIT

H-index:	38	Uptrend:	2.03	Diversity:	1.5
#Papers:	281	Activity:	7.24	Sociability:	415.27
#Citations:	6403	Longevity:	27	More Statistics...	

Dr. C. Lee Giles is the avid Reese Professor at the College of Information Sciences and Technology at the Pennsylva State versity, University Park, PA. He is also Professor of Computer Science and Engineerin sor of Supply Chain and Information Systems, and Director of the Intelligent Systems Research L irects the CiteSeerX project and codirects the ChemXSeer project at Penn State. He has been ed with Columbia University, the University of Maryland, University of Pennsylvania, Princeton Un , the University of Pisa and the University of Trento

EDIT BIO

Research Interest

Neural Networks,digital library,Web Applications EDIT

Show Temporal Interests (Do you want to see the change of his/her research interests?)

See Others:

S. Lawrence Follow
Coauthor-Count: 43, H-index: 18

Prasenjit Mitra Follow
Coauthor-Count: 56, H-index: 17

David Pennock Follow
Coauthor-Count: 16, H-index: 18

Gary Flake Follow
Coauthor-Count: 11, H-index: 7

Expertise:
Web Mining (32)
Mobile Robot / Hybrid Control (28)
Digital library / Information Access (28)
Information Retrieval / Probabilistic Indexing (25)
Data mining (23)
Data Mining / Query Processing (20)

Conference:
JCDL (27),WWW (21),CIKM (21),NIPS (13),SIGIR (7),ICDM (5)

Figure 2: Research interests are extracted from fields "Research Interest" and "Expertise" of each author profile on ArnetMiner.

3.2 Author Database from Citeseer^X

Citeseer^X hosts over 1.5 million scholarly documents. The author information (names, affiliations, lists of publications, etc.) is extracted from the documents as part of the metadata extraction. The authors are disambiguated using a machine learning based disambiguation algorithm proposed by Treeratpituk et al.[14]. We obtain a database of 307,262 authors from 1,077,513 documents.

3.3 Author Research Interests from Arnetminer

Obtaining the research interests. ArnetMiner catalogs over 1,636,804 computer science researchers. We crawl all the profile pages on ArnetMiner, and extract researcher names, research interests, and lists of publications. The research interests are extracted from the fields "Research Interest" and "Expertise" as displayed in Figure 2.

Mapping Citeseer^X authors to ArnetMiner authors. We map each author in the Citeseer^X database to an author from ArnetMiner using the following algorithm:

STEP1 For each Citeseer^X author, find the list of ArnetMiner authors who have the same names.

STEP2 If more than one ArnetMiner authors have the same name, compute the TF-IDF scores between the Citeseer^X author's publication titles (concatenated into a single chunk of text) and the publication titles of each ArnetMiner author.

STEP3 Map the Citeseer^X author to the ArnetMiner author whose the publication title similarity score is the highest.

We run the above algorithm and find 156,520 Citeseer^X authors (50.94%) can be mapped to ArnetMiner authors.

3.4 Extracting Topics using WikipediaMiner

Our schemes require that both the query and the profile is a set of weighted topics, and the topics must be from the topic taxonomy. Since both the research interests extracted from ArnetMiner and queries can be any free-form text, we need to first translate these items into topics. We use WikipediaMiner for this purpose. WikipediaMiner toolkit has the ability to annotate a document with Wikipedia topics. We hence use the tool to translate a research interest item (in form of a key phrase) into topics. We build a user profile by collecting the topics translated from the research interest items. To translate a query into topics, we follow the similar approach. If the query is merely a short text, we feed the whole query to the annotator and collect the annotating topics. If the query is a large document (like the one we use for our experiment), we first segment the query into sentences using LingPipe sentence extraction tool[5], then feed each sentence to the annotator (the annotator cannot process a large text.).

4. THE SCHEMES

4.1 Topic Similarity Function

The topic similarity function $TS(t_q, t_a, t_b)$ is an atomic function that computes the similarity between two topics t_a and t_b, given a query topic t_q. The function consults the topic taxonomy, then outputs a similarity score between 0 and 1. Recall that our topic taxonomy is represented as a direct acyclic graph (DAG) where each node is a topic and each directed edge denotes sub-topic relationship. We define $paths(t_{start}, t_{end})$ as a set of paths in the topic taxonomy, each of which starts from topic t_{start} and ends at topic t_{end}.

The shortest path $SP(t_{start}, t_{end})$ is a shortest path from topic t_{start} to topic t_{end} in the topic taxonomy, or a single node t_{start} if $paths(t_{start}, t_{end}) = \emptyset$. Since the topic taxonomy is large, and hence infeasible to compute the shortest paths in real time, we pre-compute the shortest path between every pair of topics and store the pre-computation results in a database for quick look-ups. If there are more than one shortest paths between a pair of topics, the first one found will be used. We pre-compute the shortest paths among the 758,336 topics, using Dijkstra's algorithm. The process took roughly 10 days to complete, producing 139,736,685 shortest path entries.

Let $LCP(t_q, t_a, t_b)$ be the longest common path between $SP(t_q, t_a)$ and $SP(t_q, t_b)$. The length of a path is represented by the number of nodes. Now, we define our topic similarity function $TS(t_q, t_a, t_b)$ as following:

$$TS(t_q, t_a, t_b) = \frac{|LCP(t_q, t_a, t_b)|}{min(|SP(t_q, t_a)|, |SP(t_q, t_b)|)} \tag{11}$$

As an intuitive example behind Equation 11, suppose t_q = `Sport`, t_a = `Tennis`, and t_b = `Squash`. Further assume that $SP(t_q, t_a)$ = `Sport->Racket_Sport->Tennis` and $SP(t_q, t_b)$ = `Sport->Racket_Sport->Squash`. Then it follows that `LCP (Sport, Tennis, Squash)` $= 2/3 =$ `0.67`, which corresponds to the intuition that tennis and squash are partially similar in the sense that they both are racket sports.

4.2 Profile Similarity Schemes

Family	Scheme Name	Acronym
Topic Overlap	User Uniform Overlap	UUO
	User Weighted Overlap	UWO
Summation	User Weighted Sum, Query Weighted	UWS-QW
	User Weighted Sum, Query Uniform	UWS-QU
	User Uniform Sum, Query Weighted	UUS-QW
	User Uniform Sum, Query Uniform	UUS-QU
Maximization	User Weighted Max, Query Weighted	UWM-QW
	User Weighted Max, Query Uniform	UWM-QU
	User Uniform Max, Query Weighted	UUM-QW
	User Uniform Max, Query Uniform	UUM-QU

Table 1: The 10 proposed query-dependent profile similarity schemes, divided into 3 families: Topic Overlap, Summation, and Maximization.

[5] http://alias-i.com/lingpipe/

$$\textbf{ProfileSim}_{UUO}(Q, P_A, P_B) = \frac{1}{U_U} \cdot \sum_{\substack{<t_a,w_a> \\ \in P_A}} \sum_{\substack{<t_b,w_b> \\ \in P_B}} \begin{cases} TS(t_q, t_a, t_b) & ; if\ t_a = t_b \\ 0 & ; Otherwise \end{cases} \tag{1}$$

$$\textbf{ProfileSim}_{UWO}(Q, P_A, P_B) = \frac{1}{W_U} \cdot \sum_{\substack{<t_a,w_a> \\ \in P_A}} \sum_{\substack{<t_b,w_b> \\ \in P_B}} \begin{cases} (w_a + w_b) \cdot TS(t_q, t_a, t_b) & ; if\ t_a = t_b \\ 0 & ; Otherwise \end{cases} \tag{2}$$

$$\textbf{ProfileSim}_{UWS-QW}(Q, P_A, P_B) = \frac{1}{W_Q} \cdot \sum_{\substack{<t_q,w_q> \\ \in Q}} \frac{w_q}{W_U} \cdot \left(\sum_{\substack{<t_a,w_a> \\ \in P_A}} \sum_{\substack{<t_b,w_b> \\ \in P_B}} (w_a + w_b) \cdot TS(t_q, t_a, t_b) \right) \tag{3}$$

$$\textbf{ProfileSim}_{UWS-QU}(Q, P_A, P_B) = \frac{1}{U_Q} \cdot \sum_{\substack{<t_q,w_q> \\ \in Q}} \frac{1}{W_U} \cdot \left(\sum_{\substack{<t_a,w_a> \\ \in P_A}} \sum_{\substack{<t_b,w_b> \\ \in P_B}} (w_a + w_b) \cdot TS(t_q, t_a, t_b) \right) \tag{4}$$

$$\textbf{ProfileSim}_{UUS-QW}(Q, P_A, P_B) = \frac{1}{W_Q} \cdot \sum_{\substack{<t_q,w_q> \\ \in Q}} \frac{w_q}{U_U} \cdot \left(\sum_{\substack{<t_a,w_a> \\ \in P_A}} \sum_{\substack{<t_b,w_b> \\ \in P_B}} TS(t_q, t_a, t_b) \right) \tag{5}$$

$$\textbf{ProfileSim}_{UUS-QU}(Q, P_A, P_B) = \frac{1}{U_Q} \cdot \sum_{\substack{<t_q,w_q> \\ \in Q}} \frac{1}{U_U} \cdot \left(\sum_{\substack{<t_a,w_a> \\ \in P_A}} \sum_{\substack{<t_b,w_b> \\ \in P_B}} TS(t_q, t_a, t_b) \right) \tag{6}$$

$$\textbf{ProfileSim}_{UWM-QW}(Q, P_A, P_B) = \frac{1}{W_Q} \cdot \sum_{\substack{<t_q,w_q> \\ \in Q}} \frac{w_q}{M_U} \cdot \left(\underset{\substack{<t_a,w_a>\in P_A, \\ <t_b,w_b>\in P_B}}{max} \{(w_a + w_b) \cdot TS(t_q, t_a, t_b)\} \right) \tag{7}$$

$$\textbf{ProfileSim}_{UWM-QU}(Q, P_A, P_B) = \frac{1}{U_Q} \cdot \sum_{\substack{<t_q,w_q> \\ \in Q}} \frac{1}{M_U} \cdot \left(\underset{\substack{<t_a,w_a>\in P_A, \\ <t_b,w_b>\in P_B}}{max} \{(w_a + w_b) \cdot TS(t_q, t_a, t_b)\} \right) \tag{8}$$

$$\textbf{ProfileSim}_{UUM-QW}(Q, P_A, P_B) = \frac{1}{W_Q} \cdot \sum_{\substack{<t_q,w_q> \\ \in Q}} w_q \cdot \left(\underset{\substack{<t_a,w_a>\in P_A, \\ <t_b,w_b>\in P_B}}{max} \{TS(t_q, t_a, t_b)\} \right) \tag{9}$$

$$\textbf{ProfileSim}_{UUM-QU}(Q, P_A, P_B) = \frac{1}{U_Q} \cdot \sum_{\substack{<t_q,w_q> \\ \in Q}} \left(\underset{\substack{<t_a,w_a>\in P_A, \\ <t_b,w_b>\in P_B}}{max} \{TS(t_q, t_a, t_b)\} \right) \tag{10}$$

Figure 3: The mathematical descriptions of the 10 proposed query-dependent profile similarity schemes

We propose 10 query-dependent profile similarity schemes, divided into three families: topic-overlap based, summation based, and maximization based. In essence, all the schemes compute **ProfileSim**(Q, P_A, P_B), the similarity between the two profiles P_A and P_B given a query Q, where $P_A = \{< t_{a1}, w_{a1} >, ..., < t_{am}, w_{am} >\}$, $P_B = \{< t_{b1}, w_{b1} >, ..., < t_{bn}, w_{bn} >\}$, and $Q = \{< t_{q1}, w_{q1} >, ..., < t_{qk}, w_{qk} >\}$. The similarity scores are real number between 0 and 1. We list the 10 schemes in Table 1 along with their formal mathematical descriptions in Figure 3. The constants U_U, U_Q, W_U, W_Q, and M_U are defined below:

$$U_U = |\{t_{a1}, ..., t_{am}\} \cup \{t_{b1}, ..., t_{bn}\}|\ ,\ U_Q = |Q|,$$

$$W_U = \sum_{\substack{<t_a,w_a> \\ \in P_A}} w_a + \sum_{\substack{<t_b,w_b> \\ \in P_B}} w_b,\ W_Q = \sum_{\substack{<t_q,w_q> \\ \in Q}} w_q,$$

$$M_U = max\{w_{a1}, ..., w_{am}\} + max\{w_{b1}, ..., w_{bn}\}$$

5. ANECDOTAL RESULTS

We evaluate our schemes on a sample set of 34 authors selected from 9 different computer science disciplines. We list all the 34 authors in Table 2 along with their affiliations.

We choose an author from each of the 9 disciplines. For each of the chosen 9 authors, we compute the profile similarity score against all the 34 authors, using the "TextTiling"[7] paper as the query. Table 3 lists the top 20 topics extracted from the query. We expect to see strong similarities among authors from the same disciplines. Moreover, we expect to see highly pronounced similarities among authors from the information retrieval field since the query is identified to be most related to such discipline. Figure 4 shows the profile similarity scores computed using the 10 schemes. Each heatmap has 9×34 colored square grids representing the profile similarity levels. $Grid_{ij}$ displays the similarity between the authors in row i and column j. The intensity of the "blue" color correlates to the level of profile similarity (i.e. dark blue means very similar, light blue means similar, green means slightly similar, and white means independent).

The topic overlap based schemes (UUO and UWO) give correct results. The dark blue grids tend to form a diagonal line across the heatmaps, implying high profile similarities among authors within the same research areas. However, the similarity levels are very strict–the heatmaps display only

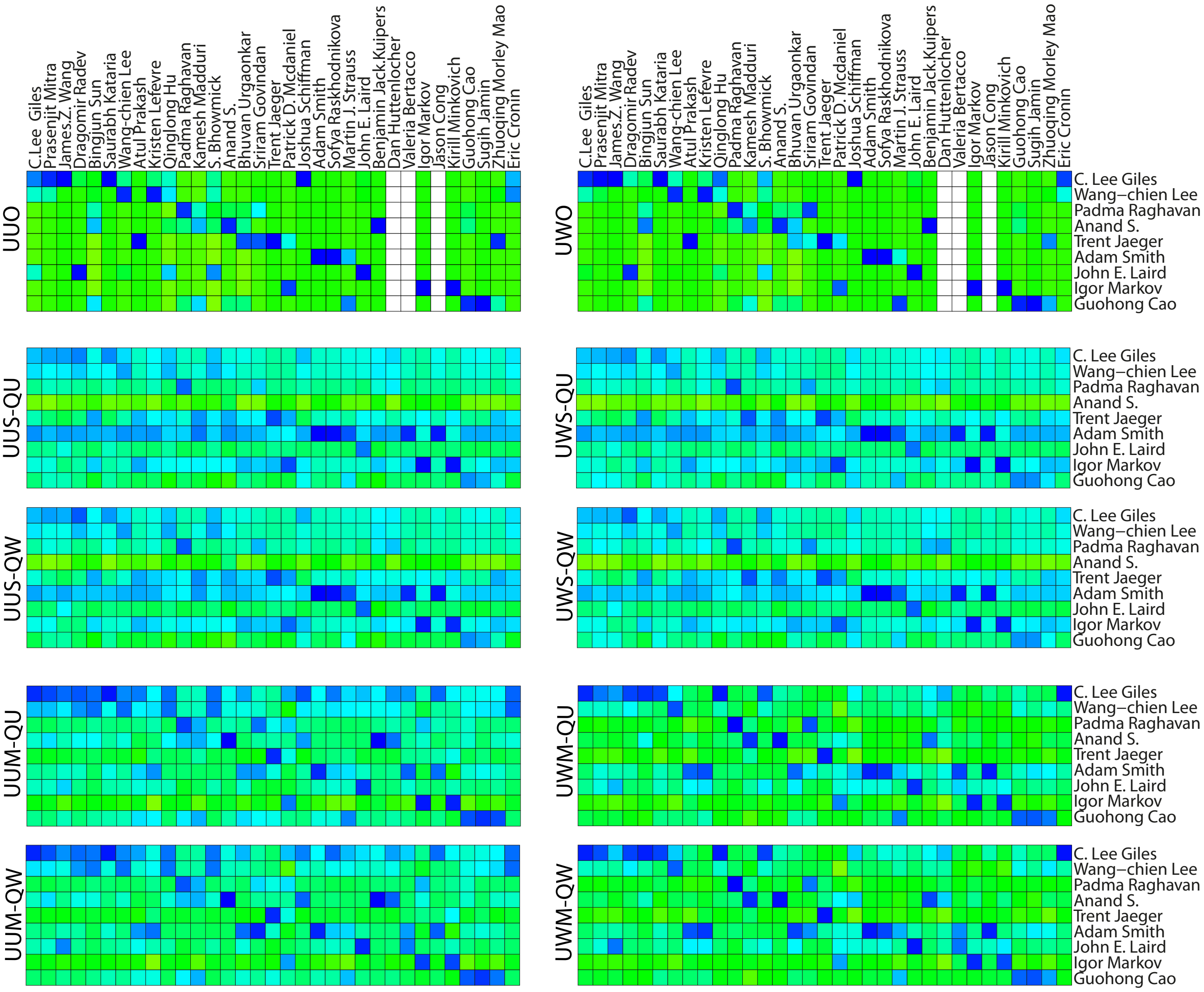


Figure 4: Anecdotal results comparing the 10 profile similarity schemes, using an information retrieval article ("TextTiling") as the query.

either dark blue grids or green (even white) grids. These high contrasts are expected since the topic overlap based schemes are not able to capture partial similarities.

The summation based schemes ($UUS-QU$, $UUS-QW$, $UWS-QU$, and $UWS-QW$) are able to compute partial similarities. However, these schemes do not yield accurate results. First, the profile similarities are not distinctive across the disciplines–the heatmaps show light blue grids spreading all over. Second, sometimes self-similarity levels are inferior to the similarities against others, which is not intuitive. For example, the similarities between C. Lee Giles and himself are even less than the similarities between C. Lee Giles and Bingjun Sun.

The maximization based schemes ($UUM-QU$, $UUM-QW$, $UWM-QU$, and $UWM-QW$) yield both correct and more accurate results than the other two families. Especially, the $UWM-QU$ and $UWM-QW$ schemes show promising diagonal blue patterns across the heatmaps. Furthermore, the profile similarities between C. Lee Giles, who is the representative of IR discipline, and the other authors in IR field (i.e. Prasenjit Mitra, James Z. Wang, Bingjun Sun, and Saurabh Kataria) are highly prominent compared to authors from other disciplines. This is expected since the query that we use is a publication from the IR field.

6. POTENTIAL APPLICATIONS

In this section, we briefly describe 2 potential applications of our proposed profile similarity schemes.

Document Ranking. In social media such as Facebook and Google+, oftentimes one would want to see posts composed by people with similar backgrounds or interests. Fortunately, users in such social networks tend to already have profiles describing their interests (music, movies, sports, etc.) and backgrounds (education, jobs, places visited, etc.). The profile similarity can then be calculated between the query issuer and other users. The scores can then be propagated to documents composed by the users which can be combined with other measures to rank documents.

Citation Recommendation. Current work in citation recommendation is primarily content-based. Topic models

Field	Name	Affiliation
Info. Retrieval	C. Lee Giles	Penn State University
	Prasenjit Mitra	Penn State University
	James Z. Wang	Penn State University
	Dragomir Radev	University of Michigan
	Bingjun Sun	Penn State University
	Saurabh Kataria	Penn State University
Databases	Wang-chien Lee	Penn State University
	Atul Prakash	University of Michigan
	Kristen Lefevre	University of Michigan
	Qinglong Hu	Creighton University
Scientific Computation	Padma Raghavan	Penn State University
	Kamesh Madduri	Penn State University
	S. Bhowmick	Nanyang Technological University
Computer Systems	Anand Sivasubramaniam	Penn State University
	Bhuvan Urgaonkar	Penn State University
	Sriram Govindan	Penn State University
Security	Trent Jaeger	Penn State University
	Patrick D. Mcdaniel	Penn State University
	Joshua Schiffman	Penn State University
Theory	Adam Smith	Penn State University
	Sofya Raskhodnikova	Penn State University
	Martin J. Strauss	University of Michigan
Artificial Intelligence	John E. Laird	University of Michigan
	Benjamin Jack Kuipers	University of Michigan
	Dan Huttenlocher	Cornell University
CAD/VLSI	Valeria Bertacco	University of Michigan
	Igor Markov	University of Michigan
	Jason Cong	University of California-LA
	Kirill Minkovich	University of California-LA
Networks	Guohong Cao	Penn State University
	Sugih Jamin	University of Michigan
	Zhuoqing Morley	University of Michigan
	Eric Cronin	University of Pennsylvania
	Changlei Liu	Penn State University

Table 2: 34 authors from 9 computer science disciplines are selected for the experiment.

Topic	**Weight**	**Topic**	**Weight**
History of mining	0.076	Data management	0.028
Mining	0.076	HCI	0.026
Data mining	0.054	Information Age	0.025
Formal sciences	0.049	Knowledge representation	0.020
Data analysis	0.048	Data modeling	0.018
Machine learning	0.046	DB management systems	0.017
Cybernetics	0.035	Research methods	0.016
Learning	0.029	Information retrieval	0.013
Nat. language processing	0.029	Library science	0.012
World Wide Web	0.027	Metadata	0.012

Table 3: Top 20 topics and weights extracted from the query (text content from [7]) using WikipediaMiner annotation tool.

are popularly used for this task where the generative models attempt to capture the document and link generation process of citation recommendation[9]. To the best of our knowledge, current models for citation recommendation do not use any information regarding the "issuer" of the query, that is, the author of the document looking for citations. In addition, current models do not model the implicit preference of the authors while adding citations. The citation behavior of authors is influenced both by the relevance of the document being cited as well as the background information of the authors of documents to be cited. Authors are inclined to cite other authors who have the same research interests and backgrounds. With such knowledge, we can use author profile similarity along with their connections in the co-authorship network to predict citations.

7. CONCLUSIONS AND FUTURE WORKS

We proposed 10 taxonomy-based query-dependent schemes for computing profile similarities. Each query and profile is defined as a set of weighted topics. The schemes are divided into three families: topic overlap based, summation based, and maximization based. The anecdotal results show that the maximization based schemes, especially $UWM - QU$ and $UWM - QW$, yield most accurate results as they are able to capture partial similarity between two topics.

We also invest our efforts harvesting resources such as the topic taxonomy from Wikipedia, the high quality list of authors from Citeseer^X, and the author research interests from ArnetMiner. We make all the resources in our research available upon request.

Despite the encouraging preliminary results, there are many aspects of our work that need improvement. Even though we pre-compute the shortest paths between all the pairs of topics in the topic taxonomy for quick look-ups, it would still take 2-3 minutes to compute the profile similarity between two profiles. This slow performance prevents the schemes from becoming scalable. Since a user profile contains 13 topics and a document query contains 700 topics on average, each computation would involve $700 \times 13 \times 13 = 118,300$ database reads, which is the cause of slow computation. A quick remedy to this problem is to choose only top topics from queries and profiles, but the results may not be as accurate as using the whole topics. Another effective solution would be to use caching, which we plan to implement into our system.

Furthermore, we believe that the anecdotal results that we provide are still insufficient to completely verify our proposed schemes. Hence we plan to equip the schemes into applications such as document search/ranking, citation recommendation, and expertise search so that we can perform more extensive evaluations on published data sets.

8. REFERENCES

[1] *mediawiki.org/wiki/Manual : Page_table.*

[2] *mediawiki.org/wiki/Manual : Categorylinks_table.*

[3] H.-H. Chen, L. Gou, X. Zhang, and C. L. Giles. Capturing missing edges in social networks using vertex similarity. In *Proceedings of the sixth international conference on Knowledge capture*, K-CAP '11, pages 195–196, New York, NY, USA, 2011. ACM.

[4] H.-H. Chen, L. Gou, X. Zhang, and C. L. Giles. Collabseer: a search engine for collaboration discovery. In *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries*, JCDL '11, pages 231–240, New York, NY, USA, 2011. ACM.

[5] S. D. Gollapalli, P. Mitra, and C. L. Giles. Ranking authors in digital libraries. In *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries*, JCDL '11, pages 251–254, New York, NY, USA, 2011. ACM.

[6] S. D. Gollapalli, P. Mitra, and C. L. Giles. Similar researcher search in academic environments. In *Proceedings of the 12th ACM/IEEE-CS joint conference on Digital Libraries*, JCDL '12, pages 167–170, New York, NY, USA, 2012. ACM.

[7] M. A. Hearst. TextTiling: segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64, 1997.

[8] G. Jeh and J. Widom. Simrank: a measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, pages 538–543, New York, NY, USA, 2002. ACM.

[9] S. Kataria, P. Mitra, and S. Bhatia. Utilizing context in generative bayesian models for linked corpus. In *In AAAI*, 2010.

[10] J. M. Kleinberg. Hubs, authorities, and communities. *ACM Computing Surveys*, 31(4es):5–es, 1999.

[11] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical Report 1999-66, Stanford Digital Library Technologies Project, 1998.

[12] P.-N. Tan, M. Steinbach, and V. Kumar. Introduction to Data Mining. *Journal of School Psychology*, 19(1):51–56, 2005.

[13] J. Tang and J. Zhang. ArnetMiner : Extraction and Mining of Academic Social Networks. *Architecture*, pages 990–998, 2008.

[14] P. Treeratpituk and C. L. Giles. Disambiguating authors in academic publications using random forests. In *Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries*, JCDL '09, pages 39–48, New York, NY, USA, 2009. ACM.