Ranking Experts using Author-Document-Topic graphs

Sujatha Das G.
Computer Science and Engineering
The Pennsylvania State University
University Park, PA 16802 USA
gsdas@cse.psu.edu

Prasenjit Mitra, C. Lee Giles
Information Sciences and Technology
Computer Science and Engineering
The Pennsylvania State University
University Park, PA 16802 USA
{pmitra,giles}@ist.psu.edu

ABSTRACT

Expert search or recommendation involves the retrieval of people (experts) in response to a query and on occasion, a given set of constraints. In this paper, we address expert recommendation in academic domains that are different from web and intranet environments studied in TREC. We propose and study graph-based models for expertise retrieval with the objective of enabling search using either a topic (e.g. "Information Extraction") or a name (e.g. "Bruce Croft"). We show that graph-based ranking schemes despite being "generic" perform on par with expert ranking models specific to topic-based and name-based querying.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval Models

Keywords

PageRank, Author-Document-Topic graphs, Expert Search, Similar Expert Finding

1. MOTIVATION

Entity search and retrieval where the goal is to retrieve "objects" (such as cars, books, people) in response to user queries is an emerging research interest in the Information Retrieval community. In particular, expert finding where the goal is to rank people with expertise (experts) in response to a topic query was well-studied in the TREC community[1]. Similarly, the list completion tracks in TREC and INEX[2] competitions address similar entity finding or exemplar-based search for the general domain. In these setups, the evidence for expertise is derived based on webpages or documents on a intranet. Documents in the academic domain are different from webpages in terms of their type (e.g. homepages, publications), structure (e.g. abstract, sections), associated metadata (e.g. venue, authors) and connections (e.g. citations). In this paper, we focus on expert (researcher/author) search in academic domains in response to both topic queries (e.g. "Information Retrieval") and name queries (e.g. "Bruce Croft").

In academic domains, an expert search system that allows queries based on topics and expert names has several potential applications. Consider the following use-cases: The program chair of a conference is desirous of selecting a panel of researchers for the "Information Retrieval" track. A potential list of PC members can be obtained by firing an appropriate topic query to an expert search system. On the other hand, consider a student applying to graduate schools who is interested in working with researchers like "Bruce Croft".

In this paper, we refer to the use case, where the input is a "topic" as *expert finding* or *topic-based search* whereas *similar expert finding* or *name-based search* is used to refer to cases when an example expert name is specified as the input. For the example in the previous paragraph, "Bruce Croft" might be associated with several other expertise areas apart from "Information Retrieval". Thus the results of these two queries need not be the same. Our focus in this paper is to explore models that are **generic** in that they allow ranking of experts in response to both topic and name queries. This is different from previous research where different models are studied for handling the two query scenarios.

Contributions and Organization: We study two graph-based models for ranking experts in response to topic-based and name-based queries. The first model is an extension of PageRank for graphs having multiple edge-types and was proposed by us previously for expert finding [9]. We show that the PageRank-based model can be used effectively for ranking experts in response to name queries as well as topic queries by constructing an appropriate query-specific graph in each case (Figure 2). The second set of scoring models is based on modeling the underlying corpus using a weighted, undirected, tripartite graph representing the Authors, Documents and Topics in the corpus (Figure 1). In contrast with the PageRank-based models that score nodes depending on their **structural connections** with other nodes in the graph, the ADT models, are designed to capture the **content-based similarity** between nodes via edges to the "topic" nodes in the graph. Indeed, nodes in ADT graphs are scored with respect to each other based on the association strengths of paths connecting them.

We study the performance of our proposed ranking models and baselines on two datasets. The first dataset is based on

[1]http://trec.nist.gov

[2]https://inex.mmci.uni-saarland.de/about.html

JCDL'13, July 22–26, 2013, Indianapolis, Indiana, USA.

ArnetMiner[3] and CiteSeer[4] and represents the retrieval environments of academic domains we seek to model where as the second dataset, the UvT collection from Tilburg university[5] provides a sparse retrieval environment [3]. Our experimental results indicate that despite being "generic" enough to handle both the querying scenarios, graph-based models rank experts on par with other query-specific models.

Section 1 described the problem addressed in this paper along with a summary of contributions. Previous research that is closely related to our problems is presented in Section 2. Details of our ranking models and a description of the baselines used in comparison experiments are described in Section 3. Section 4 provides descriptions of our datasets, experimental setup and evaluation. Finally, we conclude in Section 5 with some directions for future work.

2. RELATED WORK

The participants of the Enterprise track of TREC studied expert finding in context of enterprise data on the W3C collection. Balog, et al. proposed probabilistic models for expertise profiling and expert finding in context of sparse data environments such as webpages pertaining to research institutes and universities where the documents are more structured and relatively noise-free [2, 3].

Relevance propagation models on author-document graphs were studied by Serdyukov, et al. [25], whereas topic models were used for the same task by several researchers [12, 13, 26]. For bibliographic data, Deng, et al. showed that complicated topic models only provide slight benefits compared to probablistic models with appropriate priors [12]. Simpler voting models and vector-space representations were also studied for the expert finding task [11, 21]. In view of their simplicity and competitive performance we choose the probabilistic models proposed by Deng, et al. for bibliographic domains as the baseline method for the expert finding task in our experiments.

The list-completion tasks in TREC and INEX address the similar-entity finding task in the general domain. The proceedings of these competitions discuss various approaches for handling this task. In contrast to our problem, the input queries in these systems include a query topic description with examples of entities. The participating systems need to extract the relation between the example entities and the topic description and propose entities that hold a similar relation with the topic description, as part of the answer. For academic domains, "similar expert finding" is slightly simpler in that the objective is to find researchers that are similar to a queried researcher.

Preliminary models for "similar expert finding" were previously studied by Balog and Rijke [5] and Das, et al [10]. Balog and Rijke studied similar expert finding on the TREC data using the relations a candidate expert has with other experts, documents and terms. Hofmann, et al. considered the contextual factors such as organizational setup and combined them with content-based retrieval scores to find similar experts within an organization [17]. Das, et al. studied the same problem for academic domains and proposed several models for computing researcher profiles and similarity between researchers using these profiles.

[3]http://arnetminer.org
[4]http://citeseerx.ist.psu.edu
[5]http://ilk.uvt.nl/uvt-expert-collection

Chen, et al. presented CollabSeer that uses the structure of the co-author network to predict research collaborators in academic environments [8]. Xu, et al. use a two-layer network model that combines co-author network and researcher-concept network for making researcher recommendations [28]. Our approach targets the prediction of researchers with similar expertise profiles based on content they generate and not necessarily "co-authors". Thus while co-authorship provides evidence of similarity, we wish to design models that are capable of exploiting other sources of evidence, such as content similarity, citation behavior and so on. As an additional goal, we seek models that support both name-based and topic-based querying.

3. RANKING MODELS

Most recent works in text and document analysis adopt the view of a document as a mixture of a small number of topics. Indeed, models like Latent Dirichlet Allocation (LDA) and probabilistic latent semantic analysis (pLSA) target the extraction of abstract concepts or topics given a collection of documents [18, 6]. These models also enable the expression of a document in the corpus in terms of its topic proportion vector that corresponds to a low-dimensional representation of the document. Given a set of documents, authored by an expert, generative distribution on topics and terms can be estimated for that expert. LDA was effectively used to model scientific documents and their authors previously [13, 23].

Although modeling of authors in terms of their topic distribution is intuitive, previous work that used author similarity based on their topical profiles did not yield good performance for similar expert finding [10]. For expert retrieval in response to topic queries, LDA-style models were shown to only yield marginal benefits over simpler probabilistic counterparts [12]. *Are there alternate ways to compute similarity between authors while retaining the topical aspect of document representation?* We seek to address this question via the ADT representation (next section). We start by expressing documents in terms of their topics using content modeling tools like LDA. Next, the author-document and document-topic associations are represented via edges in a graph and paths within this graph are used to measure similarity between any pair of nodes in the graph.

3.1 The ADT tripartite graph

Let T represent the set of topics[6] associated with a document collection, D. Intuitively, an expert on a topic, $t \in T$ would have authored documents related to t and other closely-related topics. Similarly, if an author, a has expertise on a topic $t \in T$, authors similar to a could be expected to write about t and topics related to t. The associations between documents and their authors and documents and their topics can be represented by a weighted tri-partite graph as follows: Let $G = (V, E)$ represent such a graph where the vertex set, $V = A \cup D \cup T$ is the union of author, A, document, D and topic nodes, T. Edges between A and D reflect the authorship relation between documents and au-

[6]Note: In this paper, we use the term 'topic' in two senses. The first is in the context of the ADT graph and refers to the topics or concepts as extracted by tools such as LDA We also use the term in the sense of topic-queries (such as "information extraction") in context of expert finding.

thors whereas edges between D and T reflect the topical association of documents.

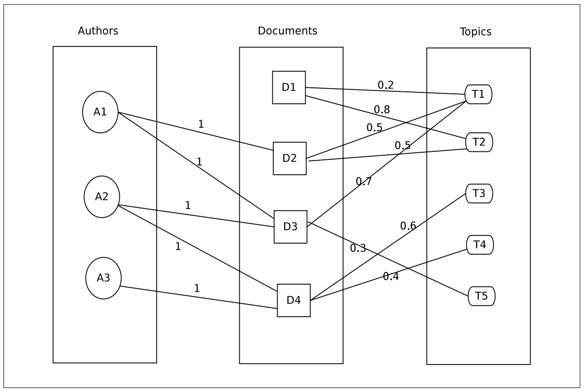

Figure 1: An example Author-Document-Topic (ADT) graph

Weights assigned to the edges in ADT capture the association strength between two nodes. For instance, in our experiments, we assign a uniform weight of 1 to all edges between author and document nodes whereas edges between document and topic nodes are assigned weights in correspondence with the proportion of that topic in the document (a positive real number). Document modeling tools such PLSI and LDA can be used for estimating these topic proportions. An example ADT graph is shown in Figure 1.

We now describe how the ADT is constructed for a given query. Consider for example, a given document node, d. First, the weighted links between this document node and its associated topic nodes are added using the proportions obtained from LDA(or PLSI). Next, other document nodes related to these topics are added to the graph along with the related edges. Finally, the known author links between the document and author layers are added to obtain a document-specific graph. If a set of documents is given as input instead of a single document (for example, documents retrieved in response to a topic query), we proceed, as in the case of a single document, by incrementally building the subgraph related to each document in the set to obtain the final graph. Instead, if for a name-query, an author node, a, is specified as input we first retrieve the documents associated with a and then follow the same procedure as with that of a set of documents.

Several well-known measures exist for comparing two nodes in weighted graphs (particularly for directed graphs). For instance, if the edge weights represent distances between nodes, one can compute proximity between two nodes in terms of the shortest path between them. In the field of network analysis, researchers have studied measures for computing vertex similarity and importance scores. A recent comparison of network similarity measures for various recommendation tasks was studied by Boldi, et al [7]. In particular, vertex similarity measures in co-authorship networks were evaluated for collaborator recommendation by Chen, et al [8].

In our preliminary experiments, we found that network centrality measures such as degree centrality, PageRank and betweenness of nodes in the ADT graph are not useful for expert ranking tasks. This is not surprising since in general centrality measures capture a node's impact with respect to the overall graph structure. These measures are not very meaningful in context of the ADT graph due to its undirected nature and the heterogenity of the underlying nodes. Moreover, our focus is on estimating a node's similarity with reference to the query nodes rather than its influence at a macroscopic level. To capture this aspect, we propose measuring the similarity between two nodes in a graph in terms of strength of paths between the nodes. Let p be a path betwen nodes a and d comprising of edges such that $p = e_1 e_2 \dots e_n$. Let

$$\mathbf{sweight}(\mathbf{p}) = \sum_i weight(e_i)$$

$$\mathbf{pweight}(\mathbf{p}) = \prod_i weight(e_i)$$

Let $P(a, d)$ be the set of all paths between nodes a and d. We studied the following schemes for computing similarity between a and d:

1. **MaxPath** The similarity between two nodes is given by the path between them having the maximum association strength. If we assign weights to edges using a transformation function that assigns weights to edges that are inversely proportional to association strengths, this scheme picks the path having the shortest distance between the nodes.

$$score(a, d) = \max_{p \in P(a,d)} \mathbf{sweight}(\mathbf{p})$$

2. **SumPaths** Consider a document that is related to two topics. An author who is associated with both the topics should be assigned credit for both the topics. The SumPaths scoring method seeks to capture this intuition by aggregating scores of all paths between two nodes in the ADT graph.

$$score(a, d) = \sum_{p \in P(a,d)} \mathbf{sweight}(\mathbf{p})$$

3. **ProductPaths** This scoring method is similar to SumPaths but we use a multiplicative scheme for aggregating edge association into a path association.

$$score(a, d) = \sum_{p \in P(a,d)} \mathbf{pweight}(\mathbf{p})$$

For a running example, suppose we would like to score the node $A1$ with respect to $D1$ in Figure 1. Assume the topic proportions for the documents are given as:

$$D1 = \{0.2, 0.8, 0, 0, 0\}, D2 = \{0.5, 0.5, 0, 0, 0\}$$

$$D3 = \{0.7, 0, 0, 0, 0.3\}, D4 = \{0, 0, 0.4, 0, 0.6\}$$

There are three paths between $D1$ and $A1$, viz.

$$\{(D1, T1), (T1, D2), (D2, A1)\}$$

$$\{(D1, T2), (T2, D2), (D2, A1)\}$$

$$\{(D1, T1), (T1, D3), (D3, A1)\}$$

With the **sweight** function, these paths have weights, 1.7, 2.3 and 2.1 respectively where as with the **pweight** function, the paths are assigned scores, 0.1, 0.4 and 0.14 respectively. Therefore, the score of $A1$ w.r.t. $D1$ is 2.3 with MaxPath, 6.1 with SumPaths and 0.64 with ProductPaths.

Notice that our scoring schemes are agnostic to the choice of nodes in the sense that all nodes are treated equally and given a node of any type, other types of nodes are assigned scores purely based on the association strengths as captured by the scoring schemes described above. This aspect is a virtue of graph-based models where query objects and the objects to be scored are both nodes and the assigned scores are simply due to some property of the underlying graph (e.g. paths, degree of nodes). Although PageRank-style measures are not meaningful for ADT graphs as we mentioned previously, we can exploit importance-based measures by constructing the underlying graph differently. An extension of PageRank using query-dependent graphs with multiple edge types was previously studied by us for scoring author nodes in response to topic queries [9]. We now show that this model can also be used for similar expert finding by constructing an appropriate query-specific graph.

3.2 PageRank on typed graphs

Objects in digital libraries have nodes corresponding to different objects such as authors, papers, venues and homepages. Similarly, edges in such graphs can represent authorship association, citation links, publication-venue links and so on. First, we briefly summarize our PageRank-based model for finding experts in response to topic-based queries before discussing name-based search.

Let $G = (V, E)$ represent a directed graph where V is the set of nodes and the edges in E have types assigned to them ($t \in T$). We use distinct transition matrices ($\mathbf{P}_t$) to capture edges corresponding to each edge type, t. Each matrix is constructed to be aperiodic and irreducible such that the aggregate transition matrix obtained by a linear combination of individual transition matrices is also irreducible and aperiodic.

$$\mathbf{P} = \sum_t w_t \mathbf{P}_t \text{ where } \sum_t w_t = 1 \text{ and } \forall w_t, 0 \leq w_t \leq 1$$

By Ergodic theorem, the matrix $\mathbf{P}$ has a unique stationary (or limiting) distribution over the nodes of V and this can be obtained by computing the principal eigen vector of the transpose of $\mathbf{P}$ (for example, using the power method [24, 14]).

The PageRank vector captures the behavior of a random surfer on the underlying graph where the final score for a node represents the probability that the surfer visits that node in the limit or as time tends to infinity. In effect, this value captures the "importance" of a node in the underlying graph based on its edge connections with the other nodes. In our extended model, we start with a query-dependent graph that has edges of different types. At every node, n, the surfer chooses at random (with probability, w_t) an edge-type, t, after which she proceeds to select one of outgoing edges of type, t, uniformly at random. The mixing coefficients, w_ts are indicative of the importance of each edge-type and can be set by domain knowledge or cross-validation.

For Expert Finding in response to topic queries, the graph on which the above model is run is ensured to be topic query dependent by using the following process: The topic query is used to retrieve the top matching documents via a search engine. This set of documents is used as an initial set of nodes, from which the remaining graph is built via expansion. For instance, the set of author nodes corresponding to the retrieved documents are added via author-document links. Other document nodes can also be added, for instance via the citation edges. We refer the reader to our previous paper for further details and experiments that show that this model performs on par with the probabilistic models for expert finding [9]. Although, this model is an extension to the original PageRank algorithm, in this paper, we use the terms **PageRank** or **PR** to refer to this extended model.

The PageRank model just described has an obvious extension that enables us to score other author nodes in response to a query-author node: We construct the query-specific graph for applying PageRank as follows. The documents associated with the queried author, a, are first retrieved to form the initial set of documents. This document set is expanded by adding other documents (via citation edges or neighbors based on content similarity). Note that this expansion step is crucial to ensure we do not only capture "co-authors". However, once an initial document set is obtained, the process is similar to that of the topic-based search case. For example, consider the graph in Figure 2 and let $A3$ correspond to the query node. The documents, $D2$ and $D3$ are retrieved via their direct association with $A3$ whereas $D4$, $A2$ and $A4$ correspond to the nodes added via our expansion process. The shaded nodes comprise the query-dependent graph used for computing PageRank scores.

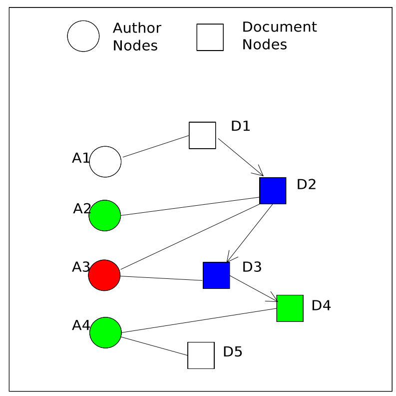

Figure 2: An example author-document graph to illustrate the PageRank-based model

There is an important difference between the ADT graph models when compared with the PageRank-based models. Due to the types of nodes and semantic meaning behind edges between nodes, similarity of nodes in the ADT models pertains to similarity based on the **topical profiles** whereas the PageRank scores capture importance based on **structural connections** in the graph. For example, an author who has written documents that are linked from other relevant documents via citation edges would be scored better in the PageRank scoring models whereas this aspect is not captured in the ADT model.

3.3 Baselines

We briefly summarize some models previously proposed for topic-based expert finding and similar expert finding.

1. **Expert Finding**: Balog, et al. [2, 3] use the estimates of $p(ca|q)$ where q is the query and ca is a candidate

for ranking experts. $p(ca|q) = \frac{p(ca,q)}{p(q)}$ and $p(ca,q)$ is defined as

$$p(ca,q) = \sum_{d \in D} p(d)p(ca,q|d) = \sum_{d \in D} p(d)p(q|d)p(ca|d,q) \quad (1)$$

D is the set of documents related to the query that a candidate ca is associated with. Assuming ca is conditionally independent of q given a document one can write $p(ca|d,q) = p(ca|d)$ and treating $p(d)$ and $p(q)$ as uniform:

$$p(ca|q) \propto \sum_{d \in D} p(q|d)p(ca|d)$$

$p(ca|d)$ is defined as $\frac{a(d,ca)}{\sum_{c' \in C} a(d,c')}$ where C is the set of all candidates and $a(d,ca)$ is the association between document d and candidate ca. The $p(q|d)$ scores for a document are estimated using language modeling.

Deng, et al. extended the probabilistic model proposed by Balog, et al. for bibliographic data [12]. The $p(ca|d)$ values were defined as $\frac{1}{n_d}$ or 0 depending on whether ca is the author of d and n_d is the number of authors for d. The prior probability, $p(d)$, was defined in terms of the number of citations that the document has. For example, $p(d) \propto ln(e + c_d)$ where c_d is the citation number for d. This probabilistic model forms a competitive baseline and complicated author-topic models were shown to obtain only marginal improvements for topic-based expert finding in bibliographic data [12].

2. **Similar Expert Finding** Adaptations of expert finding techniques were studied for similar expert finding by Balog, et al. [5] and Das, et al. [10]. In both these works, the authors in the collection are represented via their profiles constructed using the documents associated with them. Experts are recommended based on the profile similarity between two authors. We implemented these methods and chose the best performing of them as baselines (Section 4). We found that simple similarity based on TFIDF vector representation of the author profiles out-performed more complicated techniques. We briefly describe the best performing models here.

   (a) **Okapi BM25**: Given an author for whom similar experts are to be recommended, the TFIDF vector of the author is treated as a term query and the profiles of the remaining authors in the dataset are scored using the Okapi BM25 ranking function given by:

   $$\sum_{w \in s_1} IDF(w) * \frac{tf(w,s_2) * (k_1 + 1)}{tf(w,s_2) + k_1 * (1 - b + b.\frac{|D|}{avgdl}}$$

   In the above formula, $IDF(w)$ refers to the inverse document frequency of the word, a measure of rareness of the word computed as:

   $$IDF(w) = log\frac{N - N(w) + 0.5}{N(w) + 0.5}.$$

   N is the total number of profiles in the collection, $N(w)$, the number of profiles containing w and $tf(w,s_2)$, the number of times, the term w appears in the profile of e_2. The parameter k_1 is typically set to a value between $[1.2, 2]$ whereas b is typically set to 0.75 in this formula in absence of other information. Additional details on this formula and parameter settings can be found in Jones, et al. [19].

   (b) **Trace-based Similarity**: The author profiles are represented in terms of normalized TFIDF vectors of dimensionality and using the **Relevance Model** [15] studied by He, et al. the the similarity between two profiles, q and d is computed using the formula: $Rel(q,d) = (q.d)^2$.

4. EXPERIMENTS

To the best of our knowledge, no standard datasets exist for evaluating the similar expert finding task in the academic domain. However, sets of topic queries and lists of experts (traditionally referred to as qrels in TREC) for each topic are available for evaluating topic-based expert finding from ArnetMiner and Tilburg university.

4.1 Datasets

1. The ArnetMiner (**AM**) dataset provides topic+experts lists previously used by Tang, et al [27] and Deng, et al. [12] for studying topic-based expert finding in academic domains. We mapped these researcher names to the author names in CiteSeer using exact string matching. To obtain a suitable corpus for modeling the researchers, we collected a subset of document abstracts from CiteSeer by matching venues of documents with the keywords from venue names listed on Wikipedia[7]. The topic queries were used to select a subset of suitable venues in this list. We only considered authors having at least three papers in the CiteSeer collection and documents related to these authors are obtained to form the corpus.

2. The **UvT** collection was made available via the Webwijs system developed at Tilburg University. This collection contains information on UvT employees who are involved in research or teaching along with their homepages, research profiles, publications and course pages. The list of topics+experts available with this collection was used to study expert finding and profiling tasks in sparse environments by Balog, et al [3, 4]. We only considered pages in this collection that are in English.

Both the AM and the UvT datasets have 'topic' queries and manually-identified lists of authors with expertise for each topic. For evaluating similar expert finding, we need the list of experts that are similar to a given expert. We created test datasets for this purpose as follows: for a given topic query, from the set of experts listed with the query, we randomly choose one of the experts as the "name query". The other experts in the set comprise the similar researchers (or the 'gold' list) for this query. Due to the manner of construction, our 'gold' lists are in fact conditioned on the topic. Despite this dependence, we think that this dataset can be used for comparing the relative performance across ranking models. We restrict ourselves to topics that have at least five experts

[7] http://en.wikipedia.org/wiki/List_of_computer_science_conferences

Name	Description	Corpus Size	Total Authors	Queries	QRels Size
AM	ArnetMiner/CiteSeer	103838	27108	13	901
UvT	The UvT Collection	19127	1168	203	1751

Table 1: Summary of datasets used

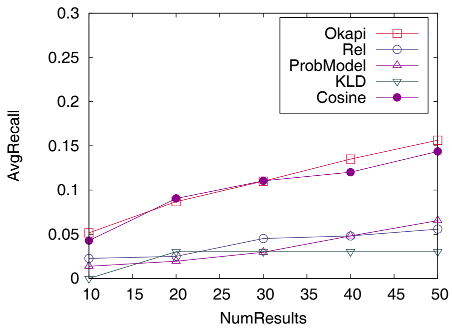

Figure 3: Comparison of baselines for Name-based search (AM)

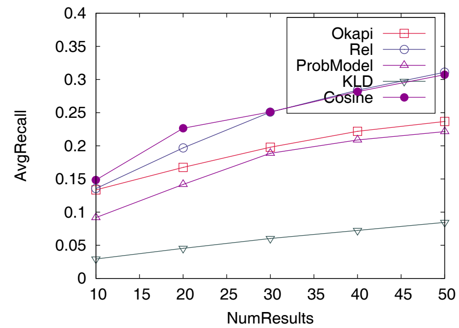

Figure 4: Comparison of baselines for Name-based search (UvT)

listed with them while forming our test set. A summary description of the datasets thus obtained is given in Table 1.

From Table 1, it can be noticed that the AM and UvT datasets correspond to rather different operating environments. The number of documents and authors and documents per author is higher in the CiteSeer/ArnetMiner although known expert-qrels are less in number. Documents in CiteSeer represent research publications and have citation links between them. In contrast, the UvT collection is sparse in that the documents mostly comprise webpages (homepages, research descriptions etc.). The number of documents per expert is limited and very few experts have more than ten documents associated with them (Table 2).

NumDocs->	1	3	5	10	20
AM/CSX	n/a	8153	2994	717	167
UvT	384	65	40	22	5

Table 2: Each entry represents the number of authors having a given number of documents associated with them. For example, the number of authors in UvT having only one document is 384

For ADT models the underlying document collection is represented in terms of topics. The "number of topics" comprises a tunable parameter for most document modeling tools like pLSA or LDA. This number is typically chosen to maximize the "likelihood" or "perplexity" of a held-out set of documents in the corpus [16]. However Azzopardi et al. showed that low perplexity representations do not necessarily result in high precision/recall for retrieval problems [1]. For this reason, we tune the "number of topics" parameter separately for each task on each dataset by using a *held-out* set of queries. In each dataset, 20% of the queries (randomly-selected) comprise the held-out set whereas the other 80% of the queries are used for evaluation and comparisons across models (Tables 5 and 6).

We present sample topic-terms for both the CiteSeer and UvT document collections in Tables 3 and 4 respectively. These topics clearly highlight the difference in the two document collections. The CiteSeer collection corresponds to documents in Computer Science and related areas and consequently the identified topics are closely related in the area. In contrast, the document collection in UvT pertains to a broader set of topics. We contend that the retrieval performance is typically better on the UvT collection since it is easier to discriminate between expertise areas corresponding to the authors.

4.2 Evaluation Setup and Metrics

The document collections in both datasets were indexed using the search engine, Indri[8]. Indri uses language modeling techniques for ranking documents in response to topic queries. We evaluate the performance of our models at different number of retrieved results ($k = 10, 20, 30, 40, 50$). In our expert finding experiments, we set the size of document set retrieved in response to the topic queries to 100 per query. The document-topic associations were obtained by running the LDA (Latent Dirichlet Allocation) implementation provided as part of Mallet[9].

We measure the performance of our proposed and baseline models using the recall, mean average precision (MAP), mean reciprocal rank (MRR) measures [22]. Let R_q represent the set of known experts for a given test query, q. If S represents the set of recommendations made by the expert retrieval system for q, we compute recall and precision for q as:

$$Recall = \frac{|S \cap R_q|}{|R_q|}$$

$$Precision = \frac{|S \cap R_q|}{|S|}$$

Average precision (AvgP) refers to the average precision with S after each relevant document is retrieved whereas MAP (mean average precision) aggregates the average precision value over all the queries (Q) to to provide a single

[8]http://www.lemurproject.org/indri/

[9]http://mallet.cs.umass.edu/

305	distribution probability random distributions show number size model independent expected uniform rate average
409	knowledge learning domain reasoning system case problem acquisition machine task expert solving base process learn
448	model models bayesian probability gaussian mixture distribution estimation likelihood maximum parameters probabilistic
302	management distributed applications system systems service application support requirements dynamic services computing
414	query queries database data databases relational processing optimization evaluation join sql efficient execution support
66	mobile devices computing wireless location users device environment user access environments services network ubiquitous
408	computational complexity based algorithm paper proposed efficient algorithms cost techniques efficiency advantage reduced
109	learning training classification data supervised labeled set approach labels learn examples class task unlabeled unsupervised
439	mining data discovery patterns association rules knowledge databases database rule frequent discover large discovering
342	semantic ontology web ontologies knowledge abstract domain semantics concepts rdf language describe resources metadata

Table 3: Sample Topics from the CiteSeer corpus

99	estimation statistics probability regression model statistical distribution estimators methods multivariate variables
98	lines prior summary top half reflects implication patterns trends greater numbers wide variety continued portion
90	index cluster clusters space target ranking coming multi group collected clustering included mixed retrieved entry
89	mind important made sense relation make arguments common consists remarks full case interpretation existence view
87	markets industrial journal firms organization competition economics collusion oligopoly consistency letters market
86	ethics law moral ethical social legal morality politics human society theory state philosophy ideals political care
83	republic europe poland czech hungary eastern state german west east central russia french case government ten
78	asia regions areas india africa rural agricultural urban historical america agriculture spread southern cities
72	criminal crime law justice police european investigation court victims prosecution enforcement victim crimes drug
71	face brain related expressions facial cognitive emotion affective emotional expression neuroscience emotions perception

Table 4: Sample Topics from the UvT collection

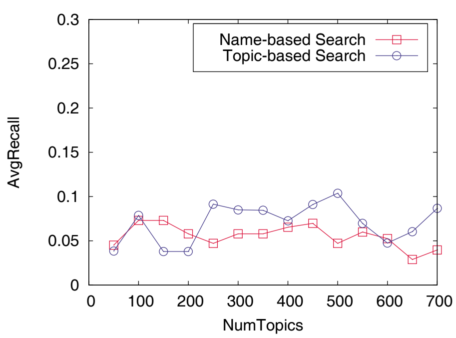

Figure 5: ADT performance variation with number of topics (AM)

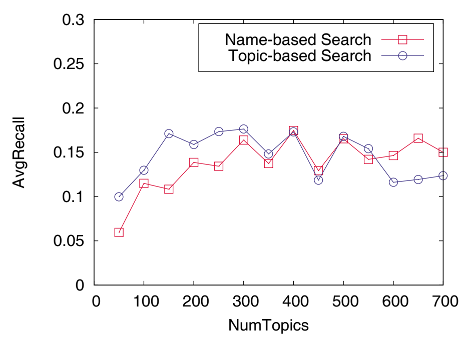

Figure 6: ADT performance variation with number of topics (UvT)

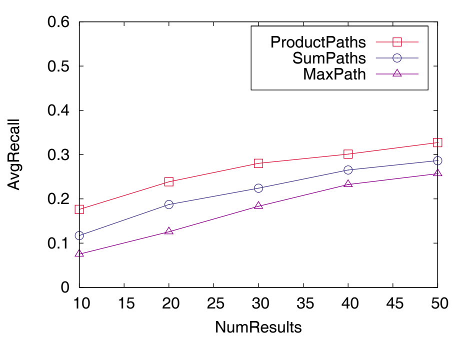

Figure 7: ADT methods for Topic-based search (UvT)

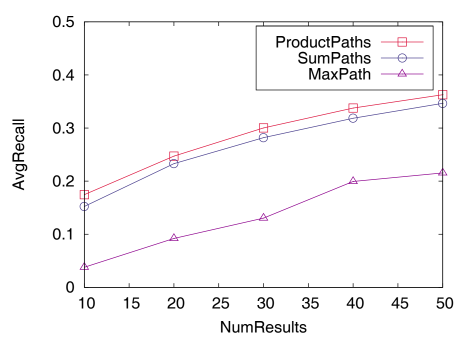

Figure 8: ADT methods for Name-based search (UvT)

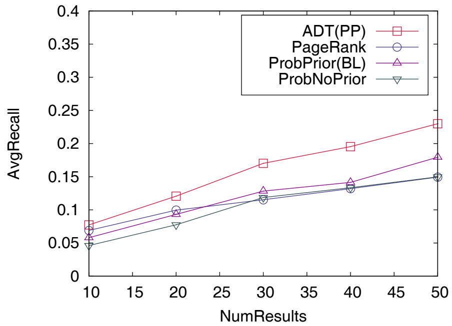

Figure 9: Recall@K for Topic-based search(AM)

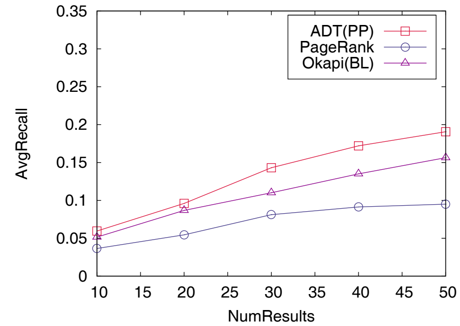

Figure 10: Recall@K for Name-based search(AM)

measure for precision as:

$$MAP = \frac{\sum_{q=1}^{Q} AvgP(q)}{|Q|}$$

The mean reciprocal rank, in contrast, values the rank at which the first correct answer is found for every query in Q and is computed using the following formula:

$$MRR = \frac{1}{|Q|} \sum_{q=1}^{Q} \frac{1}{rank(q)}$$

In the above formula, $rank(q)$ refers to the rank of the first relevant expert found for a query, q. Queries for which no relevant experts are found are considered to contribute 0 to the formula. When only the top-k predictions in S are considered for computing these measures, we append $@k$ while referring to them (e.g. Recall@10). We argue that for expert finding, recall is a more meaningful measure since it captures the number of correct experts in the gold lists that are retrieved by a method. Therefore, we use recall as the criterion for choosing the number of topics or the appropriate baselines to compare with.

4.3 Results and Observations

The performance of ADT models on the held-out queries for different number of topics is shown in Figures 5 and 6. As can be seen in these figures, "number of topics" for modeling the documents in the corpus forms a crucial parameter that affects the performance of the ADT models. Moreover, the best setting for number of topics is task-dependent. We found $numTopics = 400$ works well for both the tasks on the UvT dataset whereas for the AM dataset, $numTopics = 450, 500$ are the best settings for name-based and topic-based search respectively.

Our next set of experiments compares the different scoring schemes proposed for ADT models in Section 3 on the AM dataset for both topic-based search and name-based search (Figures 11 and 12). The SumPaths and ProductPaths clearly out-perform the MaxPath scheme that assigns scores based on the best-path connecting two nodes. For our tasks, it appears important to accumulate the scores along various paths between a given set of nodes, a feature captured by both SumPaths and ProductPaths scoring schemes. For comparison with other models, we choose ProductPaths since it is slightly better than SumPaths on the held-out queries. Similar behavior is observed on the UvT dataset in Figures 7 and 8.

Next, we compare our proposed models (ADT and PR) with the state-of-the-art baselines for these tasks. Figures 9 and 10 show the recall of the models for different number of retrieved results for the ArnetMiner dataset whereas figures 13 and 14 show similar plots for the UvT dataset. As can be seen in these figures, graph-based models (PageRank or ADT) typically outperform or perform on-par with the problem-specific baseline models in terms of recall on both the datasets. The other retrieval measures are summarized in Tables 5 and 6. For the UvT collection, simpler TFIDF-based methods seem to be better at ranking the results (as captured by MRR, for instance). We used the default setting of assigning edge-type weights (w_ts) equally in our PageRank models. The citation edges were used to expand the document set in case of name-based search in the ArnetMiner dataset whereas content-similarity (top-100 neighbors) was used to expand the set for the UvT collection (Section 3.2).

Our experiments highlight results that indicate that unified models can be designed for handling both the querying scenarios in expert search without compromising on the retrieval performance. The probablistic model was chosen as baseline for topic-based search based on its competitive performance shown in previous work [12, 9]. For name-based search, since existing work on this problem is still preliminary, we evaluated all applicable models previously proposed for this problem [5, 10] on our datasets and choose the best performing models as baselines (Figures 3 and 4).

Finally, for the sake of illustration, we provide anecdotal examples in Tables 7 and 8. These lists of top-10 recommendations were retrieved using the ADT model with the ProductPaths scoring scheme. We chose popular subject areas in Computer Science as "topic queries" and known experts in these fields as "name queries". While not all entries are perfect we found upon manual examination that most recommended authors typically publish in conferences in the related subject areas (as listed on DBLP[10]). The entries which do not seem relevant to the specified query are highlighted in italics in these tables.

5. SUMMARY AND FUTURE WORK

In this paper, we presented graph-based models for enabling expert search in response to name and topic queries. Our ADT models ranks experts based on content-similarity captured via document-topic edges. In contrast, our PageRank-

[10]http://www.informatik.uni-trier.de/∼ley/db/

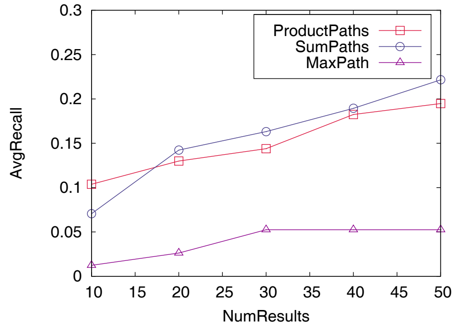

Figure 11: ADT methods for Topic-based search (AM)

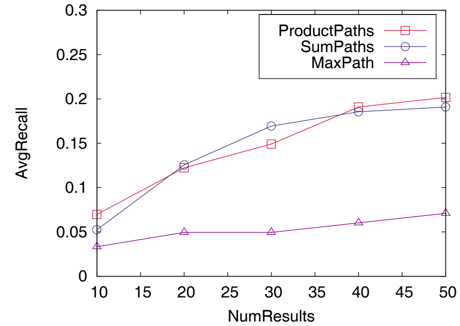

Figure 12: ADT methods for Name-based search (AM)

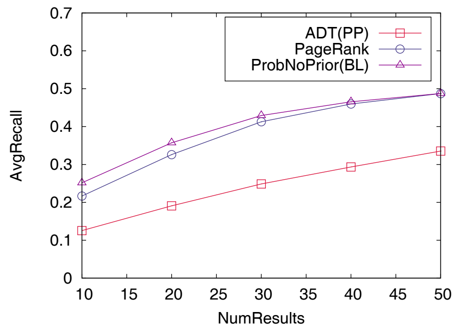

Figure 13: Recall@K for Topic-based search(UvT)

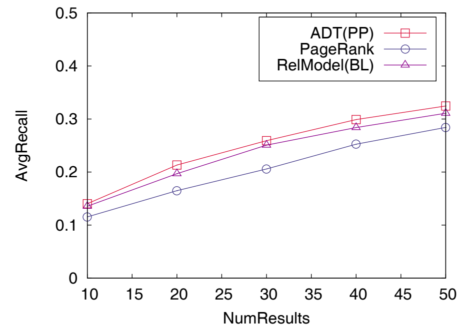

Figure 14: Recall@K for Name-based search(UvT)

ArnetMiner			
	BL(Okapi)	PR	ADT
Prec@10	0.2699	0.2100	**0.3200**
MRR@10	**0.5792**	0.3211	0.4493
MAP@10	0.1614	0.1115	**0.1999**
Prec@50	0.1720	0.1436	**0.2000**
MRR@50	**0.5908**	0.3277	0.4523
MAP@50	0.0793	0.0437	**0.1008**
UvT			
	BL(Rel)	PR	ADT
Prec@10	**0.1242**	0.1029	0.1191
MRR@10	**0.3190**	0.2673	0.2627
MAP@10	**0.0914**	0.0623	0.0875
Prec@50	0.0565	0.0529	**0.0619**
MRR@50	**0.3268**	0.2784	0.2759
MAP@50	**0.1070**	0.0729	0.1003

Table 5: Comparison Summary: Name-based Search

ArnetMiner			
	BL(Prob)	PR	ADT
Prec@10	0.3300	0.3400	**0.4300**
MRR@10	0.5009	0.6350	**0.8433**
MAP@10	0.1844	0.2097	**0.3397**
Prec@50	0.1980	0.1680	**0.2900**
MRR@50	0.5009	0.635	**0.8433**
MAP@50	0.0987	0.0851	**0.1986**
UvT			
	BL(Prob)	PR	ADT
Prec@10	**0.2158**	0.1856	0.1088
MRR@10	**0.5145**	0.4304	0.3021
MAP@10	**0.1506**	0.1245	0.0759
Prec@50	0.1245	**0.1246**	0.0598
MRR@50	**0.5201**	0.4393	0.3167
MAP@50	**0.1793**	0.1558	0.0943

Table 6: Comparison Summary: Topic-based Search

Natural Language Processing	**Machine Learning**	**Information Retrieval**	**Semantic Web**
Hermann Ney	Raymond J. Mooney	W. Bruce Croft	Ian Horrocks
Aravind K. Joshi	Vasant Honavar	Douglas W. Oard	Dieter Fensel
Raymond J. Mooney	Manuela Veloso	*Hermann Ney*	Enrico Motta
Bonnie J. Dorr	Jude Shavlik	Jamie Callan	Amit Sheth
Alex Waibel	David B. Leake	Hector Garcia-molina	Steffen Staab
Martha Palmer	Peter A. Flach	Justin Zobel	Frank Van Harmelen
Kathleen Mckeown	Pat Langley	C. Lee Giles	Stefan Decker
Udo Hahn	Yoram Singer	*Shih-fu Chang*	Rudi Studer
Alon Lavie	Ryszard S. Michalski	*Alex Waibel*	*Wolfgang Nejdl*
Bonnie Webber	Johannes Furnkranz	Jaap Kamps	*Tim Finin*

Table 7: Top-10 recommendations made by ADT models for example topic queries

Christopher D. Manning	Tom M. Mitchell	W. Bruce Croft	James Hendler
Aravind K. Joshi	Raymond J. Mooney	Douglas W. Oard	Ian Horrocks
Martha Palmer	Sebastian Thrun	Jamie Callan	Dieter Fensel
Raymond J. Mooney	Peter Stone	Justin Zobel	Amit Sheth
Timothy Baldwin	Jude Shavlik	Norbert Fuhr	Frank Van Harmelen
Bonnie J. Dorr	Vasant Honavar	Maarten De Rijke	*Wolfgang Nejdl*
John A. Carroll	Andrew Mccallum	Jaap Kamps	Enrico Motta
Ted Briscoe	Andrew G. Barto	Hector Garcia-molina	Steffen Staab
Mark Johnson	Richard S. Sutton	*Rong Jin*	*Geoffrey Fox*
Fernando Pereira	Manuela Veloso	Mounia Lalmas	Stefan Decker
Walter Daelemans	Pat Langley	*Ophir Frieder*	*Carole Goble*

Table 8: Top-10 recommendations made by ADT models for example name queries

based models rank experts using the structural connections between authors and documents and within documents. We showed via experiments that our graph-based models are capable of providing a unifed framework for ranking experts in response to both name and topic queries. In addition, we showed that these models demonstrate retrieval performance on-par with problem-specific models. Since the two proposed models are based on different types of graphs and capture different aspects of ranking (structure vs. content), a future direction to explore is the combination of model to obtain the best from both.

We are also looking to extend our expert retrieval models for panel recommendation. In contrast with topic or name-based search, recommending panels of experts imposes several problem-specific constraints. For instance, we may wish to ensure diversity in the list of experts with respect to affiliations. Metadata information available with authors (e.g. from CiteSeer or author homepages) can be used for this purpose. Similarly, personal preferences of authors with respect to each other may need to be accounted for during panel recommendation [20].

6. ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation.

7. REFERENCES

[1] L. Azzopardi, M. Girolami, and K. van Risjbergen. Investigating the relationship between language model perplexity and ir precision-recall measures. In *SIGIR*, 2003.

[2] K. Balog, L. Azzopardi, and M. de Rijke. Formal models for expert finding in enterprise corpora. In *SIGIR*, 2006.

[3] K. Balog, T. Bogers, L. Azzopardi, M. de Rijke, and A. van den Bosch. Broad expertise retrieval in sparse data environments. In *SIGIR*, 2007.

[4] K. Balog and M. de Rijke. Determining expert profiles (with an application to expert finding). In *IJCAI*, 2007.

[5] K. Balog and M. de Rijke. Finding similar experts. In *SIGIR*, 2007.

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 2003.

[7] P. Boldi, M. Rosa, and S. Vigna. Robustness of social networks: comparative results based on distance distributions. In *Social informatics*, 2011.

[8] H.-H. Chen, L. Gou, X. Zhang, and C. L. Giles. Collabseer: a search engine for collaboration discovery. In *JCDL*, 2011.

[9] S. Das, P. Mitra, and C. L. Giles. Ranking authors in digital libraries. In *JCDL*, 2011.

[10] S. Das, P. Mitra, and C. L. Giles. Similar researcher search in academic environments. In *JCDL*, 2012.

[11] G. Demartini, J. Gaugaz, and W. Nejdl. A vector space model for ranking entities and its application to expert search. In *ECIR*, 2009.

[12] H. Deng, I. King, and M. R. Lyu. Formal models for expert finding on dblp bibliography data. In *ICDM*, 2008.

[13] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 2004.

[14] T. Haveliwala, S. K. Kamvar, D. Klein, C. Manning, and G. G. Golub. Computing pagerank using power extrapolation. In *Stanford University Technical Report*, 2003.

[15] Q. He, J. Pei, D. Kifer, P. Mitra, and C. L. Giles. Context-aware citation recommendation. In *WWW*, 2010.

[16] G. Heinrich. Parameter estimation for text analysis. Technical report, 2004.

[17] K. Hofmann, K. Balog, T. Bogers, and M. de Rijke. Contextual factors for finding similar experts. *JASIST*, 2010.

[18] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR*, 1999.

[19] K. S. Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: development and comparative experiments. *Inf. Process. Manage.*, 2000.

[20] T. Lappas, K. Liu, and E. Terzi. Finding a team of experts in social networks. In *KDD*, 2009.

[21] C. Macdonald and I. Ounis. Voting for candidates: adapting data fusion techniques for an expert search task. In *CIKM*, 2006.

[22] C. D. Manning, P. Raghavan, and H. SchÃijtze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[23] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *Uncertainty in artificial intelligence*, 2004.

[24] S. M. Ross. *Introduction to Probability Models, Ninth Edition*. Academic Press, Inc., 2006.

[25] P. Serdyukov, H. Rode, and D. Hiemstra. Modeling multi-step relevance propagation for expert finding. In *CIKM*, 2008.

[26] J. Tang, R. Jin, and J. Zhang. A topic modeling approach and its integration into the random walk framework for academic search. In *ICDM*, 2008.

[27] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. Arnetminer: extraction and mining of academic social networks. In *KDD*, 2008.

[28] Y. Xu, J. Hao, R. Y. Lau, J. Ma, W. Xu, and D. Zhao. A personalized researcher recommendation approach in academic contexts: Combining social networks and semantic concepts analysis. In *PACIS*, 2010.