Learning to Diversify Expert Finding with Subtopics[⋆]

Hang Su[1], Jie Tang[2], and Wanling Hong[1]

[1] School of Software, Beihang University, China, 100191
{suhang, wanling}@sse.buaa.edu.cn
[2] Department of Computer Science and Technology, Tsinghua University, China, 100084
jietang@tsinghua.edu.cn

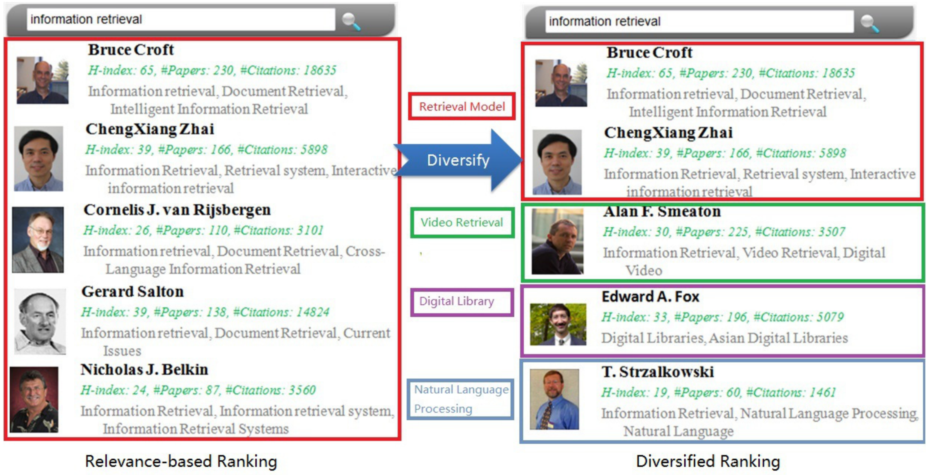
Abstract. Expert finding is concerned about finding persons who are knowledgeable on a given topic. It has many applications in enterprise search, social networks, and collaborative management. In this paper, we study the problem of diversification for expert finding. Specifically, employing an academic social network as the basis for our experiments, we aim to answer the following question: Given a query and an academic social network, how to diversify the ranking list, so that it captures the whole spectrum of relevant authors' expertise? We precisely define the problem and propose a new objective function by incorporating topic-based diversity into the relevance ranking measurement. A learning-based model is presented to solve the objective function. Our empirical study in a real system validates the effectiveness of the proposed method, which can achieve significant improvements (+15.3%-+94.6% by MAP) over alternative methods.

1 Introduction

Given a coauthor network, how to find the top-k experts for a given query q? How to diversify the ranking list so that it captures the whole spectrum of relevant authors' expertise? Expert finding has long been viewed as a challenging problem in many different domains. Despite that considerable research has been conducted to address this problem, e.g., [3, 17], the problem remains largely unsolved. Most existing works cast this problem as a web document search problem, and employ traditional relevance-based retrieval models to deal with the problem.

Expert finding is different from the web document search. When a user is looking for expertise collaborators in a domain such as "data mining", she/he does not typically mean to find *general* experts in this field. Her/his intention might be to find experts on different *aspects* (subtopics) of data mining (e.g., "association rules", "classification", "clustering", or "graph mining"). Recently, diversity already becomes a key factor to address the uncertainty and ambiguity problem in information retrieval [12, 21]. However, the diversification problem is still not well addressed for expert finding. In this paper, we try to give an explicit diversity-based objective function for expert finding, and to leverage a learning-based algorithm to improve the ranking performance.

[⋆] The work is supported by the Natural Science Foundation of China (No. 61073073) and Chinese National Key Foundation Research (No. 60933013), a special fund for Fast Sharing of Science Paper in Net Era by CSTD.



information retrieval

Bruce Croft
H-index: 65, #Papers: 230, #Citations: 18635
Information retrieval, Document Retrieval, Intelligent Information Retrieval

ChengXiang Zhai
H-index: 39, #Papers: 166, #Citations: 5898
Information Retrieval, Retrieval system, Interactive information retrieval

Cornelis J. van Rijsbergen
H-index: 26, #Papers: 110, #Citations: 3101
Information retrieval, Document Retrieval, Cross-Language Information Retrieval

Gerard Salton
H-index: 39, #Papers: 138, #Citations: 14824
Information retrieval, Document Retrieval, Current Issues

Nicholas J. Belkin
H-index: 24, #Papers: 87, #Citations: 3560
Information Retrieval, Information retrieval system, Information Retrieval Systems

information retrieval

Bruce Croft
H-index: 65, #Papers: 230, #Citations: 18635
Information retrieval, Document Retrieval, Intelligent Information Retrieval

ChengXiang Zhai
H-index: 39, #Papers: 166, #Citations: 5898
Information Retrieval, Retrieval system, Interactive information retrieval

Alan F. Smeaton
H-index: 30, #Papers: 225, #Citations: 3507
Information Retrieval, Video Retrieval, Digital Video

Edward A. Fox
H-index: 33, #Papers: 196, #Citations: 5079
Digital Libraries, Asian Digital Libraries

T. Strzalkowski
H-index: 19, #Papers: 60, #Citations: 1461
Information Retrieval, Natural Language Processing, Natural Language

Fig. 1: Illustrative example of diversified expert finding. The query is "information retrieval", and the list on the left side is obtained by language model. All the top five are mainly working on retrieval models. The right list is obtained by the proposed diversified ranking method with four subtopics (indicated by different colors).

Motivating Examples To illustrate this problem, Figure 1 gives an example of diversified expert finding. The list on the left is obtained by using language model, a state-of-the-art relevance-based ranking model [2]. We see that all the top five experts are mainly working on information retrieval models. The right list is obtained using the proposed diversified ranking method with four subtopics. The top two experts are working on information retrieval models, but the third one is working on multimedia retrieval, the fourth is about digital library, and the fifth is about information retrieval using natural language processing. The diversified ranking list is more useful in some sense: the user can quickly gain the major subtopics of the query, and could refine the query according to the subtopic that she/he is interested in. Additionally, the user can have the hint about what the other users are recently interested in, as the ranking list is obtained by learning from the user feedback (e.g., users' click data).

We aim to conduct a systematic investigation into the problem of diversifying expert finding with subtopics. The problem is non-trivial and poses a set of unique challenges. First, how to detect subtopics for a given query? Second, how to incorporate the diversity into the relevance-based ranking score? Third, how to efficiently perform the expert ranking algorithm so that it can be scaled up to handle large networks?

Contributions We show that incorporating diversity into the expert finding model can significantly improve the ranking performance (+15.3%-+94.6% in terms of MAP) compared with several alternative methods using language model, topic model and random walk. In this work, we try to make the following contributions:

- We precisely formulate the problem of diversified expert finding and define an objective function to explicitly incorporate the diversity of subtopics into the relevance ranking function.

- We present a learning-based algorithm to solve the objective function.
- We evaluate the proposed method in a real system. Experimental results validate its effectiveness.

Organization Section 2 formulates the problem. Section 3 explains the proposed method. Section 4 presents experimental results that validate the effectiveness of our methodology. Finally, Section 5 reviews related work and Section 6 concludes.

2 Problem Definition

In this section, we formulate the problem in the context of academic social network to keep things concrete, although adaption of this framework to expert finding in other social-network settings is straightforward.

Generally speaking, the input of our problem consists of (1) the results of any topic modeling such as predefined ontologies or topic cluster based on pLSI [9] or LDA [5] and (2) a social network $G = (V, E)$ and the topic model on authors V, where V is a set of authors and $E \subset V \times V$ is a set of coauthor relationships between authors. More precisely, we can define a topic distribution over each author as follows.

Topic distribution: In social networks, an author usually has interest in multiple topics. Formally, each user $v \in V$ is associated with a vector $\theta_v \in \mathbb{R}^T$ of T-dimensional topic distribution ($\sum_z \theta_{vz} = 1$). Each element θ_{vz} is the probability (i.e., $p(z|v)$) of the user on topic z.

In this way, each author can be mapped onto multiple related topics. In the meantime, for a given query q, we can also find a set of associated topics (which will be depicted in detail in §3). Based on the above concepts, the goal of our diversified expert finding is to find a list of experts for a given query such that the list can maximally cover the associated topics of the query q. Formally, we have:

Problem 1. **Diversified Expert Finding.** Given (1) a network $G = (V, E)$, (2) T-dimensional topic distribution $\theta_v \in \mathbb{R}^T$ for all authors v in V, and (3) a metric function $f(.)$, the objective of diversified expert finding for each query q is to maximize the following function:

$$\sum_{z=1}^{T} f(k|z, G, \Theta, q) \times p(z|q) \tag{1}$$

where $f(k|z, G, \Theta, q)$ measures the relevance score of top-k returned authors given topic z; we can apply a parameter τ to control the complexity of the objective function by selecting topics with larger probabilities (i.e., minimum number of topics that satisfy $\sum_z p(z|q) \geq \tau$). In an extreme case ($\tau = 1$), we consider all topics.

Please note that this is a general formulation of the problem. The relevance metric $f(k|z, G, \Theta, q)$ can be instantiated in different ways and the topic distribution can also be obtained using different algorithms. Our formulation of the diversified expert finding is very different from existing works on expert finding [3, 16, 17]. Existing works have mainly focused on finding relevant experts for a given query, but ignore the diversification over different topics. Our problem is also different from the learning-to-rank work [11, 23], where the objective is to combine different factors into a machine learning model to better rank web documents, which differs in nature from our diversified expert finding problem.

3 Model Framework

3.1 Overview

At a high level, our approach primarily consists of three steps:

- We employ an unified probabilistic model to uncover topic distributions of authors in the social network.
- We propose an objective function which incorporates the topic-based diversity into the relevance-based retrieval model.
- We present an efficient algorithm to solve the objective function.

3.2 Topic Model Initialization

In general, the topic information can be obtained in many different ways. For example, in a social network, one can use the predefined categories or user-assigned tags as the topic information. In addition, we can use statistical topic modeling [9, 5, 20] to automatically extract topics from the social networking data. In this paper, we use the author-conference-topic (ACT) model [20] to initialize the topic distribution of each user. For completeness, we give a brief introduction of the ACT model. For more details, please refer to [20].

ACT model simulates the process of writing a scientific paper using a series of probabilistic steps. In essence, the topic model uses a latent topic layer $Z = \{z_1, z_2, ..., z_T\}$ as the bridge to connect the different types of objects (authors, papers, and publication venues). More accurately, for each object it estimates a mixture of topic distribution which represents the probability of the object being associated with every topic. For example, for each author, we have a set of probabilities $\{p(z_i|a)\}$ and for each paper d, we have probabilities $\{p(z_i|d)\}$. For a given query q, we can use the obtained topic model to do inference and obtain a set of probabilities $\{p(z_i|q)\}$. Table 1 gives an example of the most relevant topics for the query "Database".

3.3 *DivLearn*: Learning to Diversify Expert Finding with Subtopics

Objective Function Without considering diversification, we can use any learning-to-rank methods [11] to learn a model for ranking experts. For example, given a training data set (e.g., users' click-through data), we could maximize normalized discounted cumulative gain (NDCG) or Mean Average Precision (MAP). In this section, we use MAP as the example in our explanation. Basically, MAP is defined as:

Table 1: Most relevant topics (i.e., with a higher $p(z|q)$) for query "Database".

Topic	$p(z\|q)$
Topic 127: Database systems	0.15
Topic 134: Gene database	0.09
Topic 106: Web database	0.07
Topic 99: XML data	0.05
Topic 192: Query processing	0.04

$$MAP(Q,k) = \frac{1}{|Q|}\sum_{j=1}^{|Q|}\frac{\sum_{i=1}^{k} Prec(a_{ji}) \times rel(a_{ji})}{\sum_{i=1}^{k} rel(a_{ji})} \tag{2}$$

where Q is a set of queries in the training data; $Prec(a_{ji})$ represents the precision value obtained for the set of top i returned experts for query q_j; $rel(a_{ji})$ is an indicator function equaling 1 if the expert a_{ji} is relevant to query q_j, 0 otherwise. The normalized inner sum denotes the average precision for the set of top k experts and the normalized outer sum denotes the average over all queries Q.

Now, we redefine the objective function based on a generalized MAP metric called MAP-Topic, which explicitly incorporates the diversity of subtopics. More specifically, given a training data set $\{(q^{(j)}, A_{q^{(j)}})\}$, where $q^{(j)} \in Q$ is query and $A_{q^{(j)}}$ is the set of related experts for query $q^{(j)}$, we can define the following objective function:

$$\mathcal{O} = \frac{1}{|Q|}\sum_{j=1}^{|Q|}\sum_{z=1}^{T} p(z|q) \times \frac{\sum_{i=1}^{k} rel(a_{ji}) \times \frac{\sum_{m=1}^{i} p(z|a_{jm})}{i}}{\sum_{i=1}^{k} p(z|a_{ji}) \times rel(a_{ji})} \tag{3}$$

where $rel(a_{ji})$ is an indicator function with a value of 1 if a_{ji} is in $A_{q^{(j)}}$, 0 otherwise.

Linear Ranking Model To instantiate the expert ranking model, we define different features. For example, for expert finding in the academic network, we define features such as the number of publications, h-index score of the author, and the language model-based relevance score. For the i-th feature, we define $\phi_i(a,q)$ as the feature value of author a to the given query q. Finally, without loss of generality, we consider the linear model to calculate the score for ranking experts, thus have

$$s(a,q) = w^T \Phi\,(a,q) = \sum_{i=1}^{N} w_i \phi_i\,(a,q) \tag{4}$$

where w_i is the weight of the i-the feature. Given a feature weight vector w, according to the objective function described above, we can calculate a value, denoted as $\mathcal{O}(w)$, to evaluate the ranking results of that model. Thus our target is to find a configuration of w to maximize $\mathcal{O}(w)$.

3.4 Model Learning

Many algorithms can be used for finding the optimal w in our model, such as hill climbing [15], gene programming(GP) [10], random walk, gradient descent [4]. For the pur-

Algorithm 1 Model learning algorithm.

```
Input: training samples $S$
Output: learned parameters $w$
Initialize $globalBestW, step$
for $circle = 1 \to loop$ do
  $w \leftarrow empiricalVector + randomVector$
  repeat
    $w_{new} \leftarrow w + step$
    if $\mathcal{O}(w_{new}) > \mathcal{O}(w)$ then
      $w \leftarrow w_{new}$
    else
      Update $step$
    end if
  until convergence
  if $\mathcal{O}(w) > \mathcal{O}(globalBestW)$ then
    $globalBestW \leftarrow w$
  end if
end for
return $globalBestW$
```

pose of simplicity and effectiveness, in this paper, we utilize the hill climbing algorithm due to its efficiency and ease of implementation. The algorithm is summarized in Algorithm 1.

Different from the original random start hill climbing algorithm which starts from pure random parameters, we add our prior knowledge $empiricalVector$ to the initialization of w, as we know some features such as BM25 will directly affect the relevance degree tends to be more important. By doing so, we could reduce the CPU time for training.

4 Experiment

We evaluate the proposed models in an online system, Arnetminer[3].

4.1 Experiment Setup

Data Sets From the system, we obtain a network consisting of 1,003,487 authors, 6,687 conferences, and 2,032,845 papers. A detailed introduction about how the academic network has been constructed can be referred to [19]. As there is no standard data sets available, and also it is difficult to create such an data sets with ground truth. For a fair evaluation, we construct a data set in the following way: First, we select a number of most frequent queries from the query log of the online system; then we remove the overly specific or lengthy queries (e.g., ‘A Convergent Solution to Subspace Learning’) and normalize similar queries (e.g., ‘Web Service’ and ‘Web Services’ to ‘Web

[3] `http://arnetminer.org`

Table 2: Statistics of selected queries. $Entropy(q) = -\sum_{i=1}^{T} p(z_i|q) \log p(z_i|q)$ measures the query's uncertainty; $\#(\tau = 0.2)$ denotes the minimum number of topics that satisfy $\sum P(z|q) \geq \tau$.

Query	Data Source Venue	Entropy	$\#(\tau = 0.2)$	$\#(\tau = 0.1)$
Data Mining	KDD 08-11	4.9	3	5
Information Retrieval	SIGIR 08-11	4.8	3	8
Software Engineering	ICSE 08-11	4.5	1	3
Machine Learning	NIPS 08-11 & ICML 08-11	4.6	2	4

Service'). Second, for each query, we identify the most relevant (top) conferences. For example, for 'Web Service', we select ICWS and for 'Information Retrieval', we select SIGIR. Then, we collect and merge PC co-chairs, area chairs, and committee members of the identified top conferences in the past four years. In this way, we obtain a list of candidates. We rank these candidates according to the appearing times, breaking ties using the h-index value [8]. Finally, we use the top ranked 100 experts as the ground truth for each query.

Topic Model Estimation For the topic model (ACT), we perform model estimation by setting the topic number as 200, i.e., $T = 200$. The topic number is determined by empirical experiments (more accurately, by minimizing the perplexity [2], a standard measure for estimating the performance of a probabilistic model, the lower the better). The topic modeling is carried out on a server running Windows 2003 with Dual-Core Intel Xeon processors (3.0 GHz) and 4GB memory. For the academic data set, it took about three hours to estimate the ACT model.

We produce some statistics for the selected queries (as shown in Table 2). $Entropy(q)$ measures the query's uncertainty and $\#(\tau = 0.2)$ denotes the minimum number of topics that satisfy $\sum P(z|q) \geq \tau$.

Feature Definition We define features to capture the observed information for ranking experts of a given query. We consider two types of features: 1) query-independent features (such as h-index, sociability, and longevity) and 2) query-dependent features (such as BM25 [13] score and language model with recency score). A detailed description of the feature definition is given in Appendix.

Evaluation Measures and Comparison Methods To quantitatively evaluate the proposed method, we consider two aspects: relevance and diversity. For the feature-based ranking, we consider six-fold cross-validation(i.e. five folds for training and the rest for testing) and evaluate the approaches in terms of Prec@5, Prec@10, Prec@15, Prec@20, and MAP. And we conduct evaluation on the entire data of the online system (including 916,946 authors, 1,558,499 papers, and 4,501 conferences). We refer to the proposed method as *DivLearn* and compare with the following methods:

RelLearn: A learning-based method. It uses the same setting (the same feature definition and the same training/test data) as that in *DivLearn*, except that it does not consider the topic diversity and directly use MAP as the objective function for learning.

Table 3: Performance for expert search approaches (%).

Approach	Prec@5	Prec@10	Prec@15	Prec@20	MAP
LM	21.1	18.3	15.6	13.6	26.3
BM25	17.8	16.7	14.8	14.7	27.1
ACT	23.3	22.8	21.5	21.4	32.3
ACT+RW	21.1	20.6	20.7	18.9	34.6
LDA	12.2	15	15.9	15.6	20.4
pLSI	21.1	21.1	19.3	18.6	31.8
RelLearn	27.8	24.4	24.8	**26.1**	35.8
DivLearn	**35.6**	**28.3**	**26.7**	25.8	**41.3**

Language Model: Language model(LM) [2] is one of the state-of-the-art approaches for information retrieval. It defines the relevance between an expert (document) and a query as a generative probability: $p(q|d) = \prod_{w \in q} p(w|d)$.

BM25 [13]: Another state-of-the-art probabilistic retrieval model for information retrieval.

pLSI: Hofmann proposes the probabilistic Latent Semantic Indexing(pLSI) model in [9]. After modeling, the probability of generating a word w from a document d can be calculated using the topic layer: $p(w|d) = \sum_{z=1}^{T} p(w|z)p(z|d)$. To learn the model, we use the EM algorithm[9].

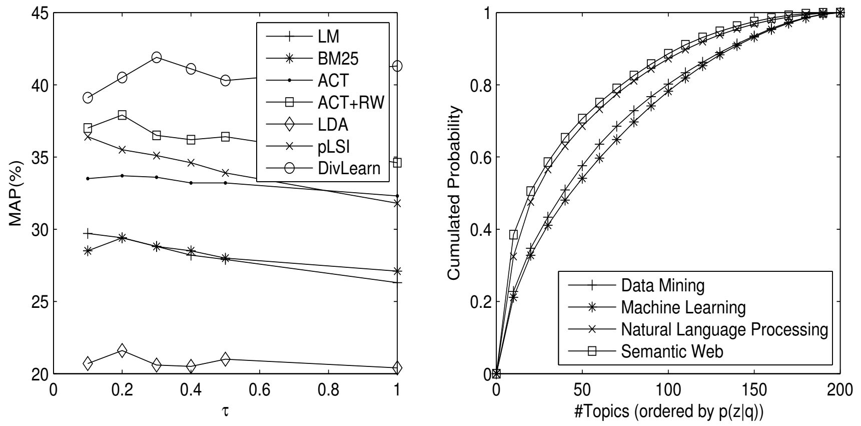
LDA: Latent Dirichlet Allocation (LDA) [5] also models documents by using a topic layer. We performed model estimation with the same setting as that for the ACT model.

ACT: ACT model is presented in §3. As the learned topics is usually general and not specific to a given query, only using it alone for modeling is too coarse for academic search [22], so the final relevance score is defined as a combination with the language model $p(q|a) = p_{ACT}(q|a) \times p_{LM}(q|a)$.

ACT+RW: A uniform academic search framework proposed in [17], which combines random walk and the ACT model together.

4.2 Performance Comparison

Table 3 lists the performance results of the different comparison methods. It can be clearly seen that our learning approach significantly outperforms the seven comparison methods. In terms of P@5, our approach achieves a +23% improvement compared with the (LDA). Comparing with the other expert finding methods, our method also results in an improvement of 8-18%. This advantage is due to that our method could combine multiple sources of evidences together. From Table 3, we can also see that the learning-based methods (both RelLearn and DivLearn) outperform the other relevance-based methods in terms of all measurements. Our DivLearn considers the diversity of topics, thus further improve the performance.



(a) Effect of threshold τ on MAP

(b) Cumulated probabilities $\sum_z p(z|q)$ for top topics of example queries

Fig. 2: Effect of topic threshold analysis

4.3 Analysis and Discussion

Now, we perform several analysis to examine the following aspects of *DivLearn*:(1) convergence property of the learning algorithm; (2) effect of different topic threshold; and (3) effect of recency impact function in Eq. 7.

Convergence Property We first study the convergence property of the learning algorithm. We trace the execution of 72 random hill climbing runs to evaluate the convergence of the model learning algorithm. On average, the number of iterations to find the optimal parameter w varies from 16 to 28. The CPU time required to perform each iteration is around 1 minute. This suggests that the learning algorithm is efficient and has a good convergence property.

Effect of Topic Threshold We conduct an experiment to see the effect of using different thresholds τ to select topics in the objective function (Eq. 3). We select the minimum number of topics with higher probabilities that statisfy $\sum_z p(z|q) \geq \tau$, then re-scale this sum to be 1 and assign 0 to other topics. Clearly, when $\tau = 1$, all topics are counted. Figure 2a shows the value of MAP of multiple methods for various τ. It shows that this metrics is consistent to a certain degree. The performance of different methods are relatively stable with different parameter setting. This could be explained by Figure 2b, which depicts the cumulated $P(z|q)$ of top n topics. As showed, for a given query, $p(z|q)$ tends to be dominated by several top related topics. Statistics in Table 2 also confirm this observation. All these observations confirm the effectiveness of the proposed method.

Effect of Recency We evaluate whether expert finding is dynamic over time. In Eq. 7, we define a combination feature of the language model score and the recency score (Func 1). Now, we qualitatively examine how different settings for the recency impact

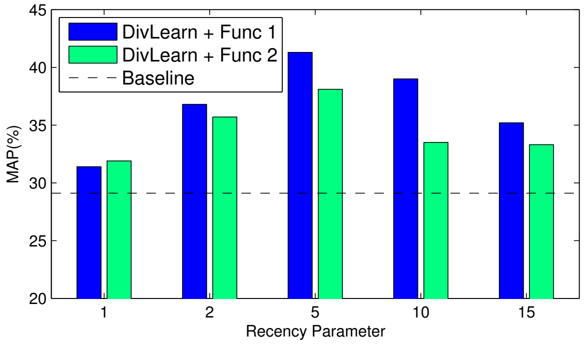


Fig. 3: MAP for different recency impact functions with different parameters.

function will affect the performance of *DivLearn*. We also compared with some other recency function with $Recency(p) = 2^{\left(\frac{\text{d.year - current year}}{\lambda}\right)}$ (Func 2) [14]. Figure 3 shows the performance of MAP with different parameter λ. The baseline denote the performance without considering recency. It shows that recency is an important factor and both impact functions perform better than the baseline which does not consider the recency. We can also see that both impact function perform best with the setting of $\lambda \simeq 5$. On average, the first impact function (Func 1, used in our approach) performs a bit better than Func 2.

5 Related work

Previous works related to our learning to diversify for expert finding with subtopics can be divided into the following three aspects: expert finding, learning to rank, search result diversification. On expert finding, [17] propose a topic level approach over heterogenous network. [3] extended language models to address the expert finding problem. TREC also provides a platform for researchers to evaluate their models[16]. [7] present a learning framework for expert finding, but only relevance is considered. Other topic model based approaches were proposed either[17].

Learning to rank aims to combining multiple sources of evidences for ranking. Liu [11] gives a survey on this topic. He categorizes the related algorithms into three groups, namely point-wise, pair-wise and list-wise. To optimize the learning target, in this paper we use an list-wise approach, which is similar to [23].

Recently, a number of works study the problem of result diversification by taking inter-document dependencies into consideration [1, 25, 6, 18]. Yue and Joachims [24] present a SVM-based approach for learning a good diversity retrieval function. For evaluation, Agrawal et al. [1] generalize classical information retrieval metrics to explicitly account for the value of diversification. Zhai et al. [25] propose a framework for evaluating retrieval different subtopics of a query topic. However, no previous work has been conducted for learning to diversify expert finding.

6 Conclusion

In this paper, we study the problem of learning to diversify expert finding results using subtopics. We formally define the problem in a supervised learning framework. An objective function is defined by explicitly incorporating topic-based diversity into the relevance based ranking model. An efficient algorithm is presented to solve the objective function. Experiment results on a real system validate the effectiveness of the proposed approach.

Learning to diversify expert finding represents a new research direction in both information retrieval and data mining. As future work, it is interesting to study how to incorporate diversity of relationships between experts into the learning process. In addition, it would be also interesting to detect user intention and to learn weights of subtopics via interactions with users.

References

1. R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *WSDM'09*, pages 5–14. ACM, 2009.
2. R. Baeza-Yates, B. Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM press New York, 1999.
3. K. Balog, L. Azzopardi, and M. de Rijke. Formal models for expert finding in enterprise corpora. In *SIGIR'06*, pages 43–50. ACM, 2006.
4. D. Bertsekas. *Nonlinear programming*. Athena Scientific Belmont, MA, 1999.
5. D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. In *NIPS'01*, pages 601–608, 2001.
6. J. G. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *SIGIR'98*, pages 335–336, 1998.
7. Y. Fang, L. Si, and A. Mathur. Ranking experts with discriminative probabilistic models. In *SIGIR 2009 Workshop on LRIR*. Citeseer, 2009.
8. J. E. Hirsch. An index to quantify an individual's scientific research output. *Proceedings of the National Academy of Sciences*, 102(46):16569–16572, 2005.
9. T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR'99*, pages 50–57. ACM, 1999.
10. J. Koza. *On the programming of computers by means of natural selection*, volume 1. MIT press, 1996.
11. T. Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, 2009.
12. F. Radlinski, P. N. Bennett, B. Carterette, and T. Joachims. Redundancy, diversity and interdependent document relevance. *SIGIR Forum*, 43:46–52, December 2009.
13. S. Robertson, S. Walker, M. Hancock-Beaulieu, M. Gatford, and A. Payne. Okapi at trec-4. In *Proceedings of TREC*, volume 4, 1995.
14. M. Roth, A. Ben-David, D. Deutscher, G. Flysher, I. Horn, A. Leichtberg, N. Leiser, Y. Matias, and R. Merom. Suggesting friends using the implicit social graph. In *KDD'10*, 2010.
15. S. Russell, P. Norvig, J. Canny, J. Malik, and D. Edwards. *Artificial intelligence: a modern approach*, volume 74. Prentice hall Englewood Cliffs, NJ, 1995.
16. I. Soboroff, A. de Vries, and N. Craswell. Overview of the trec 2006 enterprise track. In *Proceedings of TREC*. Citeseer, 2006.
17. J. Tang, R. Jin, and J. Zhang. A topic modeling approach and its integration into the random walk framework for academic search. In *ICDM'08*, pages 1055–1060, 2008.

18. J. Tang, S. Wu, B. Gao, and Y. Wan. Topic-level social network search. In *KDD'11*, pages 769–772. ACM, 2011.
19. J. Tang, L. Yao, D. Zhang, and J. Zhang. A combination approach to web user profiling. *ACM TKDD*, 5(1):1–44, 2010.
20. J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. Arnetminer: extraction and mining of academic social networks. In *KDD'08*, pages 990–998, 2008.
21. H. Tong, J. He, Z. Wen, R. Konuru, and C.-Y. Lin. Diversified ranking on large graphs: an optimization viewpoint. In *KDD'11*, pages 1028–1036, 2011.
22. X. Wei and W. Croft. Lda-based document models for ad-hoc retrieval. In *SIGIR'06*, pages 178–185. ACM, 2006.
23. J. Yeh, J. Lin, H. Ke, and W. Yang. Learning to rank for information retrieval using genetic programming. In *SIGIR 2007 Workshop on LR4IR*. Citeseer, 2007.
24. Y. Yue and T. Joachims. Predicting diverse subsets using structural svms. In *ICML'08*, pages 1224–1231. ACM, 2008.
25. C. Zhai, W. Cohen, and J. Lafferty. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *SIGIR'03*, pages 10–17. ACM, 2003.

Appendix: Feature Definition

This section depicts how we define features in our experiment. In total, we defined features of two categories: *query-independent* and *query-dependent*.

- *h-index:* h-index equals h indicates that an author has h of N papers with at least h citations each, while the left $(N-h)$ papers have at most h citations each.
- *Longevity:* Longevity reflects the length of an author's academic life. We consider the year when one author published his/her first paper as the beginning of his/her academic life and the last paper as the end year.
- *Sociability:* The score of an author's sociability is defined based on how many co-author he/she has. This score is defined as:

$$Sociability(A) = 1 + \sum_{c \in \text{A's coauthors}} ln(\#co - paper_c) \tag{5}$$

where $\#co - paper_c$ denotes the number of papers coauthored between the author and the coauthor c.
- *Language Model with Recency:* We consider the effect of recency and impact factor of conference. Thus the language model score we used for an author is redefined as:

$$LM(q|a) = \sum_{d \in \{\text{a's publications}\}} p(q|d) \times Impact(d.conference) \times Recency(d) \tag{6}$$

where $Recency(d)$ for publication d is defined as:

$$Recency(d) = \exp\left(\frac{\text{d.year - current year}}{\lambda}\right) \tag{7}$$

- *BM25 with Recency:* It defines a similar relevance score as that in Eq. 6, except that the $p(q|d)$ is obtained by BM25.