A Mixture Model for Expert Finding*

Jing Zhang, Jie Tang, Liu Liu, and Juanzi Li

Department of Computer and Technology, Tsinghua University
1-308, FIT Building, Tsinghua University, Beijing, China, 100084
{zhangjing, tangjie, ljz}@keg.cs.tsinghua.edu.cn

Abstract. This paper addresses the issue of identifying persons with expertise knowledge on a given topic. Traditional methods usually estimate the relevance between the query and the support documents of candidate experts using, for example, a language model. However, the language model lacks the ability of identifying semantic knowledge, thus results in some *right* experts cannot be found due to not occurrence of the query terms in the support documents. In this paper, we propose a mixture model based on Probabilistic Latent Semantic Analysis (PLSA) to estimate a hidden semantic theme layer between the terms and the support documents. The hidden themes are used to capture the semantic relevance between the query and the experts. We evaluate our mixture model in a real-world system, ArnetMiner[1]. Experimental results indicate that the proposed model outperforms the language models.

1 Introduction

Expert finding, aiming at answering the question: "Who are experts on topic X?", is becoming one of the biggest challenges for information management [15]. Recent years, expert finding has attracted much attention due to the rapid flourish of the Web 2.0 applications and the advancement of information retrieval technologies from the traditional document-level to the object-level [20]. Many challenging questions arise, for example, How to find the most appropriate collaborators for a project? How to find the important scientists on a research topic? How to find an expertise consultant?

Much research work has been done to deal with the challenges. For example, [2][21] propose using conventional language models for finding experts from an enterprise corpora or a domain-specific document collection. TREC has provided a common platform for researchers to empirically assess methods and techniques devised for expert finding. The task can be described as follows: given a set of documents, a list of candidate names, and a set of topics, the goal then is to find experts from the list of candidate names for each of these topics.

Previously, the language model like method or information retrieval based method is usually used for finding experts for a topic. A relevance score is calculated by

* The work is supported by the National Natural Science Foundation of China (90604025, 60703059), Chinese National Key Foundation Research and Development Plan (2007CB310803), and Chinese Young Faculty Research Funding (20070003093).

[1] http://www.arnetminer.org

T. Washio et al. (Eds.): PAKDD 2008, LNAI 5012, pp. 466–478, 2008.

combining relevance scores between the query and different support documents related to each expert candidate. Based on the combination methods, the approach can be again classified into two categories: '*composite*' and '*hybrid*'. *Composite* combines the scores of different documents by aggregation and *hybrid* integrates the scores of different support documents into a single formula (cf. Section 3 for details of the two methods). However, preliminary experiments show that simply applying these two categories of models on the task of expert finding does not achieve satisfactory results. In traditional IR models, documents are taken as the retrieval units and the content of documents are considered reliable. However, the reliability assumption is no longer valid in the expert finding context. This is because:

(1) Composite model (cf. Section 3.2.1) suffers from the limitation that all the query terms should occur in each support document.
(2) Hybrid model (cf. Section 3.2.2) is a bit more flexible. However, it still requires that all the query terms should occur in the support documents.

The language model-based methods are lexical-level and suffer from lacking semantics. A question, thus, arises: "Can we search for experts in a semantic-level?".

In this paper, we focus on the above problems. We propose a mixture model based on Probabilistic Latent Semantic Analysis (PLSA) [16] for the expert finding task. In this model, we do not model the relevance between a query and a document directly. Instead, we propose to use a hidden theme layer to model the semantic relations between the query and the support documents of candidate experts. In this way, an expert whose support documents associated with the same themes as that of a query can be ranked higher, although they may not contain the query terms. We evaluated the proposed approach in ArnetMiner system. We compared our approach with the traditional language models for expert finding. We also carried out the comparison with several existing systems. Experimental results show that our proposed approach performs better than the baseline methods and also outperforms the existing systems.

Our contributions in this paper include: (a) formalization of the expert finding problem in a semantic-level, (b) proposal of a mixture model to the problem based on Probabilistic Latent Semantic Analysis (PLSA), and (c) empirical verification of the effectiveness of the proposed approach. To the best of our knowledge, no previous work has been done on a semantic-level model for expert finding.

The rest of the paper is organized as follows. In Section 2, we formalize the task of expert finding. In Section 3, we briefly introduce the language model and propose our mixture model for expert finding. In Section 4, we give the experimental results and in Section 5, we introduce the related work. We conclude the paper in Section 6.

2 Expert Finding Description

We denote a candidate expert as e and a query as q. A general process of expert finding is to estimate the probability of a person being an expert for a given query, i.e, $P(e|q)$, and then return the experts with the highest probabilities on the top.

Based on the Bayes rule, we can obtain the following formula:

$$P(e \mid q) = \frac{P(q \mid e)P(e)}{P(q)} \xrightarrow{P(q)\ is\ uniform} P(e \mid q) \propto P(q \mid e)P(e) \quad (1)$$

where $P(q|e)$ is the generating probability of a query q given an expert e. $P(e)$ and $P(q)$ respectively denote the prior probability of an expert e and a query q. $P(q)$ is usually viewed as uniform and thus can be ignored. The probability $P(e)$ reflects the query-independent expertise. A variety of techniques can be used to compute $P(e)$, for example, we can simply use the number of one's publications to estimate the probability; more complicated, we can calculate it by using a propagation scheme like the state-of-the-art PageRank algorithm. Also, some work assumes it uniformly and only focuses on estimating the probability $P(q|e)$ using language models [2] [21].

Figure 1 shows an example of expert finding. The left part of the figure gives three queries: "semantic web", "machine learning", and "natural language processing" and the right part of the figure shows the found experts for each query.

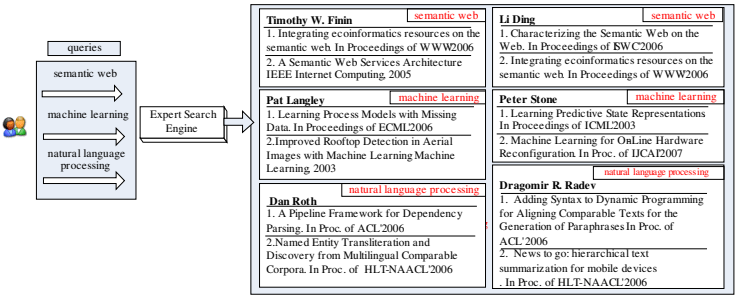

Fig. 1. An example of expert finding

3 Models for Expert Finding

In this section, we will first briefly introduce the language model and then describe several existing language models for expert finding, namely a hybrid model and a composite model. Finally, we propose a mixture model for finding experts.

3.1 Language Models for Document Retrieval

In document retrieval, language model describes the relevance between a document and a query as the generating probability of the query from the document's model:

$$P(d \mid q) \propto P(q \mid d)P(d) \tag{2}$$

For a query q, we usually assume that terms appear independently in it, thus:

$$P(q \mid d) = \prod_{t_i \in q} P(t_i \mid d) \tag{3}$$

where t_i is the i-th term in q and $P(t_i|d)$ represents the probability of generating term t_i from the language model of document d. A common method for estimating $P(t_i|d)$ is maximum likelihood estimation and Dirichlet smoothing [1], as follows:

$$P(t_i \mid d) = \lambda \cdot \frac{tf(t_i, d)}{|d|} + (1-\lambda) \cdot \frac{tf(t_i, D)}{|D|}, \qquad \lambda = \frac{|d|}{|d| + \mu} \tag{4}$$

where $|d|$ is the length of document d; $tf(t_i, d)$ is the term frequency of term t_i in d; $|D|$ is the number of documents in the document collection D; $tf(t_i, D)$ is the term frequency of term t_i in D; λ is a parameter ranging in [0, 1] and is often set based on the length of document d; μ is another parameter and is commonly set as the average document length in D.

3.2 Language Models for Expert Finding

The simplest method to apply language model for expert finding is to merge all support documents of a candidate expert together and treat them as a virtual document, then employ the language model described in Section 3.1 to estimate the relevance between the virtual document and the query. However, this model has obvious disadvantages: it cannot differentiate the contributions of different support documents. Based on the consideration, two extended language models have been proposed (we call them as composite language model and hybrid language model).

3.2.1 Composite Language Model

Let $D_e = \{d_j\}$ denotes the collection of support documents related to a candidate e. In the composite language model, each support document d_j is viewed as a unit and the estimations of all the documents of a candidate e are combined. We have:

$$P(q \mid e) = \sum_{d_j \in D_e} P(q \mid d_j) P(d_j \mid e) \tag{5}$$

The model consists of two components: 1) a document that is related to a candidate is selected with probability $P(d_j|e)$; and 2) the query q is generated from the selected document with probability $P(q|d_j)$. The former actually indicates how a document d_j characterizes the candidate e. The probability is often viewed as identical in many language modeling applications. That is, set $P(d_j|e)$ to be 1 if expert e is the author of document d_j, otherwise 0. Let $q=\{t_i\}$, the probability $P(q|d_j)$ is estimated by Equation (3) and (4) based on the independent assumption. Finally, we obtain:

$$P(q \mid e) = \sum_{d_j \in D_e} P(d_j \mid e) \prod_{t_i \in q} P(t_i \mid d_j) \tag{6}$$

We call this model as composite model because it first integrates the probability of document d_j generating each term t_i and then combines the different document models together. The nature of the composite model is that it views documents as a "hidden" variable separating the query from a candidate such that the candidate is not directly modeled. It is based on the assumption that terms are independent in d_j. Accordingly, the model emphasizes the co-occurrence of all the query terms in the same document and gives penalty to the document that does not match the whole query [2] [21]. As for the example in Figure 1, the composite model can find the two experts for the query "semantic web". However, it does not work well for the other two queries "machine learning" and "natural language processing".

3.2.2 Hybrid Language Model

The hybrid language model (cf. Equation (7)) is similar to the composite model, except that it describes each term t_i using a combination of support documents models and then uses a language model to integrate them together.

$$P(q \mid e) = \prod_{t_i \in q} \sum_{d_j \in D_e} P(t \mid d_j) P(d_j \mid e) \tag{7}$$

The two models are not equivalent mathematically since the product and the sum cannot be interchanged. The nature of the hybrid model is that it collects all terms information from all documents associated with the given candidate and models the candidate directly. It is based on the assumption that terms are independent in all support documents of e. Thus the model does not care much about the co-occurrence of the query terms in the same support document [2] [21].As for the example in figure 1, the hybrid model works well for both the queries "semantic web" and "machine learning", as the query terms appear in the support documents of experts. Unfortunately, it cannot find the two experts for "natural language processing" because it is still based on lexical-level relevance assumption.

3.3 A Mixture Model for Expert Finding

We propose a mixture model for expert finding. We assume that there is a hidden 'semantic' theme layer $\Theta=\{\theta_1, \theta_2, \ldots, \theta_k\}$ between query q and document d_j. Each hidden theme θ_m is semantically associated with multiple queries and support documents. Similarly, each support document or query is also associated with multiple themes, respectively. In this way, given a query and a support document, we do not directly model the relevance between them. Instead, we use the hidden themes associated to them as the bridge to model the relevance. More accurately, we have:

$$P(q \mid d_j) = \sum_{m=1}^{k} P(q \mid \theta_m) P(\theta_m \mid d_j) \tag{8}$$

Here, $P(q|\theta_m)$ denotes the probability of generating a query given a theme and $P(\theta_m|d)$ denotes the probability of generating a theme given a document.

We assume that a query q and a document d are conditional independent given a theme θ_m. Then the problem becomes, for each document, how to estimate the probability $P(\theta_m|d_j)$ and for each query, how to estimate the probability $P(q|\theta_m)$, called parameter estimation. Following we introduce the method for parameter estimation.

Let T as all terms occurring in the whole document collection D. Suppose there are k hidden themes. The generative process of the data set can be described as:

(1) Select a document d with probability $P(d)$;
(2) Pick a latent theme θ_m with probability $P(\theta_m|d)$;
(3) Generate a term t with probability $P(t|\theta_m)$.

As a result, we obtain an observed pair (t, d) without θ_m.

The above generative process can be expressed as a joint probability model:

$$P(t,d) = P(d)P(t \mid d), where\ P(t \mid d) = \sum_{m=1}^{k} P(t \mid \theta_m) P(\theta_m \mid d) \tag{9}$$

Equation (9) sums over all θ_m from which the observations could have been generated, which is based on the assumption that t and d are conditional independent on θ_m. We use Bayes' formula to transform Equation (9) to get its symmetric form:

$$P(t,d) = \sum_{m=1}^{k} P(t \mid \theta_m) P(d \mid \theta_m) P(\theta_m) \tag{10}$$

In order to explain the observations (t, d), we need to estimate $P(t|\theta_m)$, $P(d|\theta_m)$ and $P(\theta_m)$ by maximizing of the log-likelihood function:

$$L = \sum_{d \in D} \sum_{t \in T} n(d,t) \log \sum_{m=1}^{k} P(t \mid \theta_m) P(d \mid \theta_m) P(\theta_m) \tag{11}$$

where $n(d, t)$ denotes the co-occurrence times of d and t.

We use Expectation-Maximization (EM) algorithm [5] to estimate the maximum likelihood. The EM algorithm begins with some initial values of $P(t|\theta_m)$, $P(d|\theta_m)$, and $P(\theta_m)$ and runs an iterative process to obtain new values based on updating formulas. The update formulas contain expectation (E) step and maximization (M) step.

In E-Step, we aim to compute the posterior probability of latent theme θ_m, based on the current estimates of the parameters:

$$P(\theta_m \mid d,t) = \frac{P(t \mid \theta_m) P(d \mid \theta_m) P(\theta_m)}{\sum_{m=1}^{k} P(t \mid \theta_m) P(d \mid \theta_m) P(\theta_m)} \tag{12}$$

In M-Step, we aim to maximize the expectation of the log-likelihood of Equation (11). By introducing Lagrange multipliers and solving partial derivative, we can obtain the following equations for re-estimated parameters:

$$P(d \mid \theta_m) = \frac{\sum_{t \in T} n(d,t) P(\theta_m \mid d,t)}{\sum_{d \in D} \sum_{t \in T} n(d,t) P(\theta_m \mid d,t)} \tag{13}$$

$$P(t \mid \theta_m) = \frac{\sum_{d \in D} n(d,t) P(\theta_m \mid d,t)}{\sum_{t \in T} \sum_{d \in D} n(d,t) P(\theta_m \mid d,t)} \tag{14}$$

$$P(\theta_m) = \frac{\sum_{d \in D} \sum_{t \in T} n(d,t) P(\theta_m \mid d,t)}{\sum_{d \in D} \sum_{t \in T} n(d,t)} \tag{15}$$

The E-step and M-step run iteratively until the log-likelihood function converges to a local maximum. Then we obtain the parameters: $P(t|\theta_m)$, $P(d|\theta_m)$, and $P(\theta_m)$.

3.4 Find Experts Using the Model

We can make inferences based on the estimated probabilities. Given a query, the probability $P(q|\theta_m)$ can be estimated by

$$P(q \mid \theta_m) = \prod_{t \in q} P(t \mid \theta_m) \tag{16}$$

Then Equation (8) can be rewritten as:

$$P(q \mid d_j) = \sum_{m=1}^{k} \prod_{t_i \in q} P(t_i \mid \theta_m) P(\theta_m \mid d_j) \tag{17}$$

Therefore we obtain Equation (18) by substituting $P(q|d_j)$ into Equation (5):

$$P(q \mid e) = \sum_{d_j \in D_e} \sum_{m=1}^{k} \prod_{t_i \in q} P(t_i \mid \theta_m) P(\theta_m \mid d_j) P(d_j \mid e) \tag{18}$$

where $P(\theta_m|d_j)$ can be estimated by Bayes' formula:

$$P(\theta_m \mid d_j) = \frac{P(d_j \mid \theta_m) P(\theta_m)}{P(d_j)} \propto P(d_j \mid \theta_m) P(\theta_m) \tag{19}$$

Now, we get the probability $P(q|e)$. We can further obtain $P(e|q)$ by $P(e|q) \propto P(q|e)\ P(e)$, where $P(e)$ is often viewed as uniform in previous work such as [3]. However, we have found that final results sometimes are sensitive to the probability. In this work, we employ the propagation approach we have proposed in [25] to estimate $P(e)$. The approach is based on the social relationship analysis. The basic idea is that if a person knows many experts on a topic or if the person's name co-occurs many times with the known experts, then it is more likely that he/she is an expert on the topic. Finally we obtain $P(e|q)$ for each candidate and sort the candidates accordingly.

4 Experiments

In this section, we first introduce the experimental setting. Then we present the experimental results. Finally we give some discussions.

4.1 Experimental Setting

We evaluate the work in the context of ArnetMiner[22]. ArnetMiner contains 448,289 researchers and 725,655 publications extracted from the Web database, pages, and files. As performing PLSA on the full data collection will take an extreme long time, we created a subset of the data for evaluation purpose. Specifically, we first selected the most frequent queries from the log of ArnetMiner (by removing the specific queries or too long queries, e.g., 'A convergent solution to tensor subspace learning'). We also removed the similar queries (e.g., 'web service' v.s. 'web services'). Then we obtained seven queries: 'information extraction' (IE), 'machine learning' (ML), 'semantic web' (SW), 'natural language processing' (NLP), 'support vector machine' (SVM), 'planning' (PL), and 'intelligent agents' (IA). Next, for each query, we gathered the top 30 persons from Libra author search , Rexa authors search, and ArnetMiner[1]. We merged all the persons together by removing ambiguous names (e.g., L. Liu) and names that do not exist in ArnetMiner. Finally we got 421 person names. We collected 14,550 publications of the 421 persons from ArnetMiner as the support document collection.

For evaluation, it is difficult to find a standard data set as the ground truth. As a result, we use the method of pooled relevance judgments [8] together with human judgments. Specifically, for each query, we first pooled the top 30 results from the above three systems (Libra, Rexa, and ArnerMiner) into a single list. Then, one faculty and two graduates, from the authors' lab, provided human judgments. Assessments were carried out mainly in terms of how many publications he/she has

published, how many publications are related to the given query, how many top conference papers he/she has published, what distinguished awards he/she has been awarded. Finally, the judgment scores were averaged to construct the final ground truth. The data set is available on line.

We conducted evaluation in terms of P@5, P@10, P@20, P@30, *R*-prec, Mean Average Precision (*MAP*) and P-R curve [8] [10].

We used the language models introduced in Section 3.2 as baselines. Hereafter, we respectively call them CM and HM. For comparison purpose, we also report the results obtained by Libra and Rexa.

We implemented our proposed model (shortly MM) in two stages. In the first stage, we use PLSA algorithm (equations (12)-(15)) to estimate the probabilities $P(t|\theta_m)$, $P(d|\theta_m)$, and $P(\theta_m)$ for each document, term, and theme. Here, documents denote publications. Terms are extracted from the titles and conference names of the publications after word segmentation and stop words filtering. We empirically set the number of themes as 300 (cf. figure 3 for the effect of the number of themes). In the second stage, we rank experts using equation (18) for each query.

4.2 Experimental Results of Expert Finding

Table 1 shows the performances on the 7 queries by our proposed model, the two language models, and the two systems (Libra and Rexa). Figure 2 shows the average 11-point precision recall curves on the 7 queries for the different approaches. We see that in terms of most of the measures, the proposed model outperforms the two baseline language models. We also present top 9 example experts for "natural language processing" ranked by different approaches in Table 2.

4.3 Discussions

(1) Improvements Over Baselines. Our proposed model outperforms the two language models in terms of P@5, P@10 and *MAP*. From the PR curve, we can also see that our model outperforms the language models in most of the 11 points, which confirms the effectiveness of our approach. The proposed model can retrieve experts whose support documents do not contain the query terms but 'semantically' related to the query, therefore our approach can improve the performance significantly. For example, in Table 2, our model MM ranks higher for "Raymond J. Mooney" than the language models. This is because many of Mooney's papers do not exactly contain the query terms although they are related to "natural language processing". We rank higher for "Dan Roth" and "Dragomir R. Radev" due to the similar reason.

(2) Effect of the Number of Themes. The best number of themes is difficult to determine. In our experiment, we tried to tune the parameter to get better performance. As Figure 3 shows, the number of themes systematically varies from 10 to 100 with interval 10 and from 100 to 1000 with interval 100. In general, the best results were obtained when setting the number of themes as 300.

An intuitive explanation to Figure 3 is that when the number of theme is small, the estimated mixture model prefers to very general queries; with the number increasing,

the model prefers to specific queries. The number 300 seems to be a best balance in our setting. Table 3 show two themes with the representative words, respectively for #theme=10 and #theme=300.

(3) Language Models. We also analyze the retrieval results of two language models. From table 1, we see that for queries "SW", "IE" and "SVM", CM performs better than HM, because the word "web", "information" and "machine" may slightly drive the topic of documents drift away when using HM. For the queries of "PL", "IA", "ML", and "NLP", HM performs better than CM, due to the limitation in CM that all the query terms should co-occur in one document.

Table 1. Performances of different expert finding approaches (%)

Query	Approach	P@5	P@10	P@20	P@30	R-pre	MAP
SW	Libra	80.00	70.00	80.00	66.67	60.00	71.28
	Rexa	80.00	60.00	55.00	43.33	37.78	52.65
	CM	80.00	80.00	75.00	70.00	62.22	76.70
	HM	80.00	80.00	85.00	76.67	60.00	69.25
	MM	100.00	100.00	75.00	60.00	57.78	72.20
IE	Libra	100.00	60.00	50.00	36.67	50.00	67.76
	Rexa	60.00	60.00	45.00	36.67	45.00	51.88
	CM	80.00	70.00	65.00	56.67	65.00	73.16
	HM	80.00	70.00	60.00	56.67	60.00	71.96
	MM	100.00	70.00	60.00	56.67	60.00	75.03
SVM	Libra	60.00	30.00	25.00	30.00	32.26	37.22
	Rexa	60.00	60.00	40.00	36.67	35.48	43.75
	CM	100.00	90.00	75.00	66.67	64.52	79.47
	HM	100.00	100.00	80.00	60.00	58.06	76.61
	MM	100.00	100.00	80.00	63.33	61.29	81.56
PL	Libra	60.00	60.00	65.00	53.33	48.57	57.02
	Rexa	60.00	70.00	60.00	46.67	42.86	52.50
	CM	80.00	70.00	65.00	56.67	54.29	70.14
	HM	100.00	90.00	75.00	60.00	54.29	73.07
	MM	80.00	90.00	70.00	60.00	54.29	74.04
IA	Libra	80.00	50.00	40.00	26.67	26.67	49.63
	Rexa	60.00	40.00	35.00	40.00	40.00	43.90
	CM	80.00	70.00	60.00	53.33	53.33	70.06
	HM	100.00	80.00	65.00	60.00	60.00	78.18
	MM	100.00	100.00	70.00	50.00	50.00	82.29
ML	Libra	60.00	40.00	35.00	30.00	29.27	33.88
	Rexa	80.00	70.00	60.00	46.67	34.15	52.52
	CM	60.00	60.00	50.00	46.67	46.34	54.96
	HM	60.00	60.00	60.00	56.67	53.66	60.07
	MM	80.00	80.00	65.00	53.33	51.22	66.70
NLP	Libra	40.00	30.00	35.00	43.33	36.59	40.49
	Rexa	20.00	20.00	30.00	26.67	24.39	26.29
	CM	40.00	70.00	65.00	50.00	0.00	61.76
	HM	80.00	70.00	55.00	60.00	48.78	68.93
	MM	100.00	80.00	65.00	60.00	48.78	76.07
AVE	Libra	68.57	48.57	47.14	40.95	40.48	51.04
	Rexa	60.00	54.29	46.43	39.52	37.09	46.21
	CM	74.29	72.86	65.00	57.14	49.39	69.46
	HM	85.71	78.57	68.57	61.43	56.40	71.15
	MM	94.29	88.57	69.29	57.62	54.76	75.41

Table 2. Top 9 experts for "natural language processing" by five expert finding approaches

MM	CM	HM	Libra	Rexa
Raymond J. Mooney	Rebecca F. Bruce	Janyce Wiebe	Eric Brill	W. Addison Woods
Dan Roth	Janyce Wiebe	Michael Collins	Christopher D. Manning	Klaus Netter
Michael Collins	Veronica Dahl	Aravind K. Joshi	Adam L. Berger	Yorick Wilks
Janyce Wiebe	Robert J. Gaizauskas	Raymond J. Mooney	Stephen Della Pietra	Kavi Mahesh
Aravind K. Joshi	Kevin Humphreys	Rebecca F. Bruce	Vincent J. Della Pietra	Robert H. Baud
Rebecca F. Bruce	Aravind K. Joshi	Veronica Dahl	David D. Lewis	Kevin Humphreys
Veronica Dahl	Philippe Blache	Robert J. Gaizauskas	Kenneth Ward Church	Philippe Blache
Claire Cardie	Eric Brill	Thomas Hofmann	Hinrich Schutze	Victor Raskin
Oren Etzioni	Raymond J. Mooney	Eric Brill	Lillian Jane Lee	Lorna Balkan

(4) Decline Over Baselines. In terms of p@20, p@30 and *R*-prec, we must note that our model underperforms the two language models. The reason lies in that our model may also bring some noises when estimating the probabilities $P(t|\theta_m)$, $P(d|\theta_m)$, and $P(\theta_m)$ in the first stage.

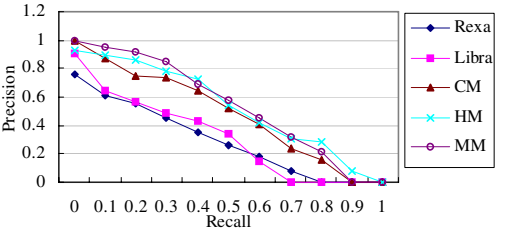

Fig. 2. Average Precision-recall curves of five expert finding approaches for 7 queries

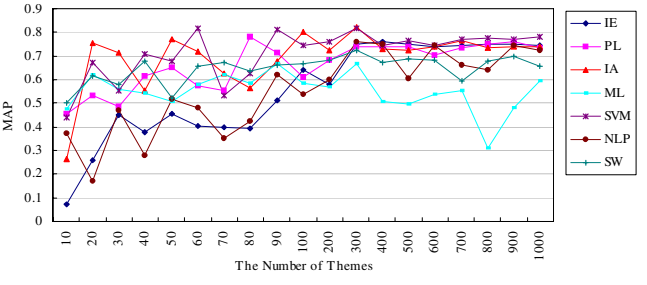

Fig. 3. The effect of the number of themes

Table 3. Example themes discovered by PLSA with #themes=10 and #themes=300. Each theme is shown with 10 representative words.

#Themes = 10										
Theme #2	information	design	framework	intelligent	ontology	management	based	semantic	systems	web
Theme #3	KDD	neural	from	text	selection	networks	Time	data	mining	using
#Themes = 300										
Theme #12	spelling	roadmap	ebl	correction	scoring	question	Directions	answering	ICGA	syntax
Theme #64	zero	variance	manifolds	predictions	principal	transformation	ICPR	matrix	clustering	words

5 Related Work

5.1 Language Model for Expert Finding

With the launch of expert finding task in TREC 2005, more and more researchers begin focusing on the research topic. Previous work for expert finding usually makes use of language models. For example, Cao et al. [9] propose a two-stage language model which combines a co-occurrence model to retrieve documents given a query, and a relevance model to find experts in those documents. Balog et al. [2] propose a model which models candidate using support documents directly and another model which is similar to the model of Cao. [3] studies the expert finding problem in a sparse data environments and proposes several advanced models based on the characteristics of the dataset. Petkova et al. analyze and compare different language models proposed for the task of finding experts [21]. They argue that all the models are probabilistically equivalent and the differences lie in the independent assumptions. As far as we know, expert finding by using latent semantic analysis has not been investigated previously.

5.2 Probabilistic Latent Semantic Analysis and Its Applications

The idea of using latent semantic structure in information retrieval traces back to [13]. They propose latent semantic analysis (LSA) method, which is mostly used in automatic indexing and information retrieval [4]. The main idea is to map data using Singular Value Decomposition (SVD) from a high-dimensional vector space repre sentation to a reduced lower representation, also called latent semantic space.

A new approach to discover latent variables is Probabilistic latent semantic analysis (PLSA) proposed by Thomas Hofmann [16]. The difference between LSA and PLSA is that the latter one is based on the likelihood principle and defines a proper generative model of the data; hence it results in a more solid statistical foundation. The core of PLSA is a statistical model called aspect model, which assumes there exists a set of hidden factors underlying the co-occurrences among two sets of objects. Expectation Maximization (EM) algorithm [5] is used to estimate the probabilities of the hidden factors generating the two sets of objects.

Probabilistic Latent Semantic Analysis has been used to solve problems in a variety of applications on account of its flexibility. Such applications include information retrieval [16], text learning and mining [6] [7] [14] [18] [24], co-citation analysis [11] [12], social annotation analysis [23], web usage mining [17] and personalize web search [19].

6 Conclusion

In this paper, we have proposed a mixture model for expert finding. We assume that there is a latent theme layers between terms and documents and employ the themes to help discover semantically related experts to a given query. A EM based algorithm has been employed for parameter estimation in the proposed model. Experimental results on real data show that our proposed model can achieve better performances than the conventional language models. As future work, we plan to investigate how to automatically determine the number of themes based on the input query.

References

[1] Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. ACM Press, New York (1999)

[2] Balog, K., Azzopardi, L., de Rijke, M.: Formal Models for Expert Finding in Enterprise Corpora. In: Proc. of SIGIR 2006, pp. 43–55 (2006)

[3] Balog, K., Bogers, T., Azzopardi, L., Rijke, M., Bosch, A.: Broad Expertise Retrieval in Sparse Data Environments. In: Proc. of SIGIR 2007, pp. 551–558 (2007)

[4] Berry, M., Dumais, S., OBrien, G.: Using Linear Algebra for Intelligent Information Retrieval. SIAM Review 37, 573–595 (1995)

[5] Bilmes, J.A.: A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Modals. Berkeley, ICSI TR-97-021 (1997)

[6] Brants, T., Chen, F., Tsochantaridis, I.: Topic-based Document Segmentation with Probabilistic Latent Semantic Analysis. In: Proc. of CIKM 2002, pp. 211–218 (2002)

[7] Brants, T., Stolle, R.: Find Similar Documents in Document Collections. In: Proc. of LREC 2002 (2002)

[8] Buckley, C., Voorhees, E.M.: Retrieval Evaluation with Incomplete Information. In: Proc. of SIGIR 2004, pp. 25–32 (2004)

[9] Cao, Y., Liu, J., Bao, S., Li, H.: Research on Expert Search at Enterprise Track of TREC (2005)

[10] Craswell, N., de Vries, A., Soboroff, I.: Overview of the Trec-2005 Enterprise Track. In: TREC 2005 Conference Notebook, pp. 199–205 (2005)

[11] Cohn, D., Chang, H.: Learning to Probabilistically Identify Authoritative Documents. In: Proc. of ICML 2000, pp. 167–174 (2000)

[12] Cohn, D., Hofmann, T.: The Missing link: A Probabilistic Model of Document Content and Hypertext Connectivity. In: Neural Information Processing Systems 13, MIT Press, Cambridge (2001)

[13] Deerwester, S., Dumais, S., Furnas, G., Landauter, T., Harshman, R.: Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science 41(6) (1990)

[14] Gaussier, E., Goutte, C., Popat, K., Chen, F.: A Hierarchical Model for Clustering and Categorizing Documents. In: Crestani, F., Girolami, M., van Rijsbergen, C.J.K. (eds.) ECIR 2002. LNCS, vol. 2291, pp. 229–247. Springer, Heidelberg (2002)

[15] Hawking, D.: Challenges in Enterprise Search. In: Proc. of the Fifteenth Conference on Australasian Database, vol. 27, pp. 15–24 (2004)

[16] Hofmann, T.: Probabilistic Latent Semantic Analysis. In: Proc. of UAI 1999 (1999)

[17] Jin, X., Zhou, Y., Mobasher, B.: Web Usage Mining based on Probabilistic Latent Semantic Analysis. In: Proc. of SIGKDD 2004, pp. 197–205 (2004)

[18] Kim, Y., Chang, J., Zhang, B.: An Empirical Study on Dimensionality Optimization in Text Mining for Linguistic Knowledge Acquisition. In: Whang, K.-Y., Jeon, J., Shim, K., Srivastava, J. (eds.) PAKDD 2003. LNCS (LNAI), vol. 2637, pp. 111–116. Springer, Heidelberg (2003)

[19] Lin, C., Xue, G., Zeng, H., Yu, Y.: Using Probabilistic Latent Semantic Analysis for Personalized Web Search. In: Zhang, Y., Tanaka, K., Yu, J.X., Wang, S., Li, M. (eds.) APWeb 2005. LNCS, vol. 3399, pp. 707–717. Springer, Heidelberg (2005)

[20] Nie, Z., Ma, Y., Shi, S., Wen, J., Ma, W.: Web Object Retrieval. In: Proc. of WWW 2007, pp. 81–90 (2007)

[21] Petkova, D., Croft, W.B.: Generalizing the Language Modeling Framework for Named Entity Retrieval. In: Proc. of SIGIR 2007 (2007)

[22] Tang, J., Zhang, D., Yao, L.: Social Network Extraction of Academic Researchers. In: Proc. of ICDM 2007, pp. 292–301 (2007)
[23] Wu, X., Zhang, L., Yu, Y.: Exploring Social Annotations for the Semantic Web. In: Proc. of WWW 2006, pp. 417–426 (2006)
[24] Zhai, C., Velivelli, A., Yu, B.: A Cross-collection Mixture Model for Comparative Text Mining. In: Proc. of SIGKDD 2004, pp. 743–748 (2004)
[25] Zhang, J., Tang, J., Li, J.: Expert Finding in a Social Network. In: Kotagiri, R., Radha Krishna, P., Mohania, M., Nantajeewarawat, E. (eds.) DASFAA 2007. LNCS, vol. 4443, pp. 1066–1069. Springer, Heidelberg (2007)