Hierarchical Language Models for Expert Finding in Enterprise Corpora

Desislava Petkova and W. Bruce Croft
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts Amherst, MA 01003
{petkova,croft}@cs.umass.edu

Abstract

Enterprise corpora contain evidence of what employees work on and therefore can be used to automatically find experts on a given topic. We present a general approach for representing the knowledge of a potential expert as a mixture of language models from associated documents. First we retrieve documents given the expert's name using a generative probabilistic technique and weight the retrieved documents according to expert-specific posterior distribution. Then we model the expert indirectly through the set of associated documents, which allows us to exploit their underlying structure and complex language features. Experiments show that our method has excellent performance on TREC 2005 expert search task and that it effectively collects and combines evidence for expertise in a heterogeneous collection.

1 Introduction

Expert finding is the task of discovering '*Who knows what*' among the employees of an organization. An expert recommender - a system which identifies people with particular skills and experience, can be a valuable management tool to promote collaboration and increase productivity by supporting knowledge sharing and transfer within and across organizations. However, expert finding systems face unique challenges that are particular to enterprise search. Information spaces within organizations are characterized by dynamic collection generation, heterogeneity due to both structured and unstructured documents in various formats, job-related task context, operational and security requirements, existence of nuanced social networks and interactions, and lack of appropriate evaluation framework [13].

A traditional approach to expert finding is to manually create, organize and control expertise information in a database [2]. However, in the context of constantly developing industrial environments, knowledge accumulation is a dynamic process, often distributed across (geographically dispersed) offices, and it can benefit from a formal, unsupervised methodology for extracting and maintaining up-to-date expertise information. In addition, there are no generic rules for formalizing expertise. Given a particular problem, the designer predefines a set of categories and subcategories for describing expertise, a framework which is too coarse or rigid for answering free keyword queries [9]. Furthermore, since the database schema is developed to serve a specific domain, task or even organization, it is hard apply it in a different context.

Recent work on automatic expert finders has formulated the problem of determining who has knowledge in a particular area as a retrieval task to rank people given a query topic. However, a standard retrieval system cannot solve this problem directly. Although enterprise corpora contain information about employees, clients, projects, meetings, etc., an expert recommender cannot find experts strictly by ranking documents. The system may begin by retrieving documents but it must then extract and process this document information in order to return a ranked list of people.

There are two principal approaches to expert modeling: query-dependent and query-independent. In both cases the expert system has to discover documents (or more generally, snippets of text) related to a person and estimate the probability of that person being an expert from the text. Commonly, a co-occurrence of the person with the query words in the same context is assumed to be evidence of expertise.

A query-dependent expert finding system ranks documents in the corpus given a query topic (the standard Information Retrieval task to retrieve documents on a given topic), and then estimates the probability of a person being an expert from the subset of retrieved documents associated with that person. For example, ExpertFinder developed at MITRE [11] first examines available sources of information (technical reports, newsletters, resumes) for documents containing the query terms. Second, it finds the employees mentioned in these documents and determines their ranking based on factors such as number of associated documents and distance between name and query terms.

A query-independent expert finding system directly models the knowledge of a person based on a set of documents associated with the candidate and estimates a probabilistic distribution of words to describe the person. An example of such a system is P@ANOPTIC Expert [7], which extracts information about employees from intranet documents and assembles expert profiles by concatenating the text of associated documents. The system then indexes these virtual 'employee documents', and given a query it retrieves a ranked list of potential experts.

We take the query-independent approach and propose a formal method for constructing query-independent expert representations, based on a statistical approach for modeling relevance. But rather than create one long document per candidate expert, we represent experts as a mixture of documents in a profile set. Our main goal is that this process is a very general entity-modeling technique which is easy to extend to take advantage of various information sources and prior knowledge about the experts, the collection or the domain. In particular, we focus on the problems of analyzing heterogeneous data, creating formal, extensible representations and answering complex relevance queries.

The remainder of this paper is organized as follows. We briefly summarize related work on expert and relevance modeling in Section 2. We describe our hierarchical language models for representing experts in Section 3 and report a series of experiments to evaluate their effectiveness in Section 4. We conclude with a discussion of our findings in Section 5.

2 Related work

Increased interest in enterprise search and its practical importance led to the introduction of the Enterprise track in the Text REtrieval Conference in 2005. The track provides a platform for working with data which reflects the interactions among the employees of an organization [6]. It includes an expert finding task with a list of potential experts to rank, a set of query topics and relevance judgments. We use the testbed provided by the Enterprise track for evaluation and comparison to other techniques.

Interestingly, last year's results showed that both the query-independent and query-dependent approaches to expert modeling can be effective: a query-independent and a query-dependent system were the two best performing systems at TREC. Fu *et al.* [8] analyze text content to extract related information and construct description files for each candidate expert. Cao *et al.* [5] propose a two-stage model which combines co-occurrence to find documents relevant to the query topic, and relevance to find experts in retrieved documents using backoff name matching.

Both methods have advantages and disadvantages [18]. A query-independent approach allows greater flexibility when identifying references to a particular expert in the text. This makes it potentially easier to address issues of named entity identification such as co-referencing and to process documents of a particular type recognizing the fact that they may reflect expertise in a characteristic way [16]. In terms of data management, profiles can be significantly smaller in size than the original corpus. On the other hand, a query-dependent approach guarantees using the most up-to-date information to model expertise. It also allows to apply advanced text modeling techniques in ranking individual documents and thus exploit structure and high-level language features, which are otherwise lost in concatenating documents to form a profile. However, aggregation of retrieval results from multiple sources poses challenges of its own and doing it at query time can lead to inefficiency.

Balog *et al.* [3] formalize and extensively compare the two methods. Their Model 1 directly models the knowledge of an expert from associated documents (the query-independent approach), and Model 2 first locates documents on the topic and then finds the associated experts (the query-dependent approach). In the reported experiments the second method performs significantly better when there are sufficiently many associated documents per candidate.

We propose an expert modeling technique which combines the two strategies. We do not explicitly create a profile document but instead form a profile set of associated documents. To estimate the probability of a candidate being an expert on a given a query topic, we analyze documents in the profile set independently and represent an expert as a mixture of the language models of associated documents.

We use language modeling to find associations between documents and experts and to estimate the strength of association. The estimation is based on the Relevance Model proposed by Lavrenko *et al.* [10], a generative modeling approach for approximating the probability distribution of terms in the relevant class of information need I. The information need is represented by a set of query terms, $Q = q_1 \ldots q_n$, which are randomly sampled from the relevance model $P(\cdot|I)$. Assuming i.i.d. sampling, the joint distribution $P(t, I)$ is estimated from a finite set of document models M.

$$\begin{aligned} P(t, I) &\approx P(t, q_1 \ldots q_n) \\ &= \sum_{M \in \mathcal{M}} P(t, q_1 \ldots q_n | M) \\ &= \sum_{M \in \mathcal{M}} P(t|M) \prod_{i=1}^{n} P(q_i|M) \end{aligned}$$

where the probabilities $P(q_i|M)$ and $P(t|M)$ are smoothed maximum likelihood estimates.

From the joint distribution, the conditional probability $P(t|I)$ can be estimated by applying Bayes formula.

$$P(t|I) = \frac{P(t,I)}{P(I)} = \frac{P(t,I)}{\sum_t P(t,I)}$$

Thus the unknown distribution $P(\cdot|I)$ is approximated given only the sample Q by computing $P(t|I)$ for every term in the vocabulary. In the context of expert modeling, the information need is a candidate expert E and the sampling space $\mathcal{M}$ is the profile set of documents associated with E.

3 Modeling experts

We estimate the topical knowledge of an expert E by a distribution over a set of words, the vocabulary V:

$$\sum_{w \in V} P(w|E) = 1$$

After building such a model and assuming that query terms are sampled independently (the '*bag-of-words*' assumption), we can use query likelihood to estimate the probability that the expert's language model generates a query Q:

$$P(Q|E) = \prod_{i=1}^{|Q|} P(q_i|E)$$

For the purpose of expert search we assume that $P(Q|E)$ reflects the degree of E being interested or involved in Q. Note that we do not define or model the concepts of '*sphere of expertise*' and '*being an expert*', and therefore we do not actually answer the question "What is the probability that the person E is an expert on the topic Q?". Instead we measure the probability that the language model describing E independently generates the words describing Q. Therefore, our system answers a weaker, but related question, while being flexible enough to model a wide scope of expertise areas.

3.1 Modeling experts as a mixture of documents

Let assume that we are provided with a list of possible experts and a set of documents. Our task is to learn about the candidates, so that given a query we can rank them by topical relevance. We propose a method for creating implicit expert representations, and a retrieval model for answering complex structured queries.

Our expert modeling approach includes the following steps:

1. For each candidate expert E, define what constitutes a reference to E, so that occurrences of E can be detected. (This is the problem of matching named entities. The original list of candidates might describe each person in alternative ways. The TREC Enterprise data, for example, specifies both the full name and at least one email address. We choose to use names because they are more flexible and we can find more associated documents.)

2. Rank and retrieve documents according to the probability $P(E|D)$. These make up the *profile set* S_E of an expert. To estimate $P(E|D)$, we apply language modeling with Dirichlet smoothing.

$$\begin{aligned} P(E = (n_1 \ldots n_k)|D) &= \prod_{i=1}^{k} P(n_i|D) \\ &= \prod_{i=1}^{k} \frac{tf_{n_i,D} + \mu P(n_i|C)}{|D| + \mu} \end{aligned}$$

   where $n_1, \ldots, n_k$ are the terms used to identify references to expert E, e.g. her first and last name.

   The size of the profile set naturally varies across experts as some people participate more actively in email discussion and other enterprise activities. However, in the experiments we report next, we retrieve the same number of documents per expert because this simplifies the model. We leave the question of automatically setting $|S_E|$ as future work.

3. For each document D in S_E, compute the posterior probability $P(D|E)$, assuming that the prior distribution is uniform.e

$$\begin{aligned} P(D|E) &= \frac{P(E|D)P(D)}{P(E)} \\ &= \frac{P(E|D)P(D)}{\sum_{D \in S_E} P(E|D)P(D)} \end{aligned}$$

   where

$$P(D) = \frac{1}{|S_E|}$$

4. Form a term distribution for E by incorporating the document model $P(t|D)$, then marginalizing.

$$\begin{aligned} P(t|E) &= \sum_D P(t|D)P(D|E) \\ &= \sum_D P(t|D)\frac{P(E|D)P(D)}{P(E)} \end{aligned} \quad (1)$$

   where $P(t|D)$ is the maximum likelihood estimate

$$P(t|D) = \frac{tf_{t,D}}{|D|}$$

That is, we represent an expert as a mixture of documents, where the mixing weights are specified by the posterior distribution $P(D|E)$.

We can compare the expert model defined in Eq. (1) with the representation used by P@NOPTIC Expert where each occurrence of a word is considered to have weight equal to 1. In contrast, we weight occurrences by $P(E|D)$, the posterior distribution of documents in the profile set S_E.

Once we have built models for all candidates, we find experts relevant to a particular topic Q by ranking the candidates according to query likelihood.

$$\begin{aligned} P(Q|E) &= \prod_{i=1}^{|Q|} P(q_i|E) \\ &= \prod_{i=1}^{|Q|} \sum_{D} P(q_i|D)P(D|E) \qquad (2) \end{aligned}$$

3.2 Building hierarchical expert models

The result of Eq. (1) is a probability distribution of words describing the context of an expert's name, where $P(\cdot|E)$ is estimated using a particular name definition and from a homogeneous collection of documents. (The assumption of homogeneity is implicit because documents are treated equivalently when building their language models.) Probability distributions estimated from different collections or alternative name definitions can be combined to build richer expert representations. For example, when working with documents in different formats, we can divide them into subcollections C, estimate an expert model $P(\cdot|E_C)$ from each subcollection and then a compute a final representation as a linear combination of several models.

$$\begin{aligned} P(t|E) &= \sum_{C \in \mathbf{C}} \lambda_C P(t|E_C) \\ \sum_{C \in \mathbf{C}} \lambda_C &= 1 \end{aligned}$$

This method of incorporating evidence for expertise can be generalized to build expert models from multiple information sources or from one source using different named entity recognition rules.

4 Experiments

In order to evaluate the flexibility and effectiveness of our expert modeling approach, we perform a series of experiments using the framework developed for the expert search task in the Enterprise track, TREC 2005. The track provides a heterogeneous document collection of 330,037 documents, a list of 1092 candidate experts with the full name and email address of each candidate, and a set of 10 training and 50 test topics.

We design our experiments to address the following research questions:

- Can advanced text modeling techniques be successfully applied to make use of complex text features?
- Can the model handle the heterogeneity naturally present in an enterprise corpus by relative weighting of subcollection evidence?
- Within a homogeneous subcollection, can the model derive further evidence of expertise by relative weighting of the structural components of documents?
- Can the model successfully leverage finding more information about an expert with the noise introduced by incorrect associations?

To run our experiments, we used the Indri search engine in the Lemur toolkit [1]. Indri integrates Bayes net retrieval model with formal statistical techniques for modeling relevance [17]. The Bayes net representation of an information need allows formulating richly structured queries: Indri powerful query language can handle phrase matching, synonyms, weighted expressions, Boolean filtering, numeric fields and proximity operators. This functionality is combined with relevance estimation based on smoothed language models. Therefore Indri can provide an efficient framework for incorporating various sources of contextual evidence.

We start by defining an expert as the phrase "*LAST-NAME FIRST-NAME*" where the two names appear unordered within a window of size 3, i.e. with at most 2 other words between them. (In Indri syntax this is expressed as *#uw3(FIRST NAME)* and we use that notation in the rest of the paper.) We build expert models using only the documents in the *web* subcollection of the W3C corpus. These settings give the baseline performance and in subsequent sections we demonstrate how the baseline can be improved by formulating complex topic queries (Section 4.1), analyzing document structure (Section 4.2), combining information sources with different intrinsic properties (Section 4.3), and combining alternative expert definitions (Section 4.4). Our goal with these experiments is not to develop new approaches for any of these specific problems. On the contrary, we apply techniques that have already been shown to improve retrieval performance for various tasks, in order to show that the expert models defined in Section 3.1 can be easily generalized and augmented by adapting various techniques developed for document retrieval.

4.1 Query expansion

The model defined in Section 3.1 can be used to answer not only simple keyword queries but also complex feature queries because we preserve documents in the profile set in their entirety, including term positions within documents. In this set of experiments, we apply two methods for automatic query expansion: pseudo-relevance feedback (to increase recall by adding terms related to the original query) and proximity constraints (to increase precision by taking advantage of dependencies between terms).

Pseudo-relevance feedback

For pseudo-relevance feedback, we implement the Relevance model proposed in [10] and discussed in Section 2:

$$P(t|I) = \frac{\sum_{D \in topDocs} P(t|D)P(I|D)P(D)}{P(I)} \tag{3}$$

where the relevance model $P(t|I)$ of information need I is computed over terms using the highest ranked N documents from an initial ranking according to $P(I|D)$. For each query topic, we construct a relevance model from the top 15 documents retrieved in an initial query and augment the original query with the 10 terms with the highest likelihood from the relevance model.

We point out the similarity between Eq. (1) and Eq. (3), which are both adaptation of the Relevance Model. To apply the Relevance Model for pseudo-relevance feedback, terms are sorted according to $P(r|I)$ and the top terms are added to the original query with weights specified by $P(r|I)$. To apply the Relevance Model for expert modeling, we build a probabilistic language model from all the terms occurring in profile set, not just the most probable ones.

Term dependency

An interesting problem in entity modeling is how to capture relationships between terms. If a query contains multiple terms, then it is important whether they co-occur in the documents forming the profile set of an expert. For example, a candidate expert can discuss t_1 in some documents and t_2 in other documents where the two sets do not overlap. This candidate should be considered less of an expert on topic $(t_1\ t_2)$ than a person who discusses both t_1 and t_2 in the same set of documents.

We implement term dependency as described by Metzler and Croft [12], using both sequential dependency and full dependency between query terms to include restrictions on terms appearing in close proximity in the text.

Results from the query expansion experiments are reported in Table 1. The evaluation measures are: mean average precision (MAP), R-precision, reciprocal rank of top relevant candidate (RR1), precision after 10 and 20 candidates retrieved (P@10 and P@20 respectively). The primary measure used to score expert search runs in the Enterprise track is MAP. We also report the number of retrieved relevant candidates (Rel-ret) because it reflects the ability of the system to successfully build representations. Both pseudo-relevance feedback and term dependency improve the mean average precision, and the improvement is compounded when the two techniques are applied together. The results show that the expert representations effectively capture both simple word features as well as higher-level language features such as phrases.

Query Model	Rel-ret	MAP	R-prec	RR1	P@10	P@20
`Q0`	585	0.2303	0.2851	0.5409	0.3820	0.3130
`Q1`	575	0.2367	0.2846	0.6107	0.3880	0.3180
`Q2`	571	0.2493	0.3091	0.5930	0.4040	0.3200
`Q3`	568	0.2551	0.3025	0.6187	0.4120	0.3190

Table 1. Results of applying different query expansion methods to the expertise topics. The query models are: baseline with no expansion (`Q0`), pseudo relevance feedback (`Q1`), term dependency (`Q2`), and feedback and term dependency combined (`Q3`).

4.2 Incorporating document structure

Emails form a considerable part of the communication in an organization and are characterized by rich internal and external structure - they are divided into fields and grouped into threads. Previous work has shown that email structure is a useful source of information in expert finding [4].

To investigate whether our model can accommodate email structure effectively, we combine evidence from the header (subject, date, to, from and cc fields), the mainbody (original text of message with reply-to and forwarded text removed), and the thread (concatenated text of messages making up the thread in which the message occurs). Similarly to the work described in [14], we define the language model of an email D_{em} as a linear combination of its three components.

$$\begin{aligned} P(t|D_{em}) &= \lambda_{hd} P(t|D_{hd}) + \lambda_{mb} P(t|D_{mb}) \\ &+ \lambda_{th} P(t|D_{th}) \end{aligned}$$

where $P(t|D_{hd}), P(t|D_{mb}), P(t|D_{th})$ are the maximum likelihood estimates from the header, mainbody and thread, respectively, and $\lambda_{hd} = 1/8, \lambda_{mb} = 5/8, \lambda_{th} = 2/8$. (We found these values to be optimal for another task in the

Enterprise track, searching for emails discussing a given topic.)

Results from the email structure experiments are reported in Table 2. For the baseline we use the entire email content (header fields and mainbody) without breaking it up into components. We exploit internal structure by weighting header and mainbody differently and external structure by adding a third component corresponding to thread text. Adding structure information improves performance and the method is easily extendable to other types of documents with well-known structure, e.g. scientific articles.

Email Structure	Rel-ret	MAP	R-prec	RR1	P@10	P@20
`NO`	419	0.1447	0.1823	0.5238	0.2780	0.2020
`YES`	433	0.1572	0.2114	0.5174	0.2980	0.2270

Table 2. Results of representing the structure of emails by combining header, mainbody and thread text.

4.3 Combining various sources of information

The W3C corpus is composed of several subcollections comprising documents of particular type. In this set of experiments, we independently build a language model from one subcollection at a time and then represent an expert as a mixture of those models. This allows us to treat each subcollection differently according to its specific intrinsic properties, e.g. when smoothing to estimate $P(E|D)$, as well as to weight the information sources, ideally taking advantage of some prior knowledge about the collections.

The W3C corpus contains an email subcollection (average length 450 words) and a web collection (average length 2000). We automatically set the Dirichlet smoothing μ parameter to the average document length, and we use the 10 training queries to experimentally determine that the optimal value for the mixing parameter λ_{www} is 0.6. Results are reported in Table 3. Although models built from the *web* subcollection significantly outperform models built from the *email* subcollection, by combining the two we achieve an even better performance, indicating that email discussion lists provide some additional information not contained in the web pages.

Collection Model	Rel-ret	MAP	R-prec	RR1	P@10	P@20
`C0`	433	0.1572	0.2114	0.5174	0.2980	0.2270
`C1`	568	0.2551	0.3025	0.6187	0.4120	0.3190
`C2`	601	0.2786	0.3220	0.6458	0.4300	0.3350

Table 3. Results of using different subcollections. We build expert models from email lists (`C0`) and web pages (`C1`), and we combine the two representations in (`C2`).

4.4 Combining expert definitions

We recognize two primary problems to be solved in expert search (and they are independent although both influence the final retrieval performance). The first problem is finding information about an expert. Failing to retrieve any documents about a candidate means that she would not be considered an expert on any query topic. The second problem is building better models for those experts about whom some information is retrieved, and we already discussed that in the previous sections. Improving on the first problem requires better ways of identifying references to a candidate.

To address this issue, we compare several expert definitions with varying strictness. We use exact match of *FIRST LAST* which is the loosest definition as many people have the same first name. We also use exact match of *LAST NAME* which is more strict but still we expect many incorrect matches. And finally we use phrases *#uwN(FIRST LAST)* with the window size N decreasing from 12 to 2, which have increasing strictness but probably do not detect many true associations, as people are not necessarily referred to with their full names, especially in emails. The number of retrieved experts and the MAP for each of these expert definitions are compared in Figures 1 and 2.

The graphs show an inverse relationship between finding more information and performance. This is a reflection of the tradeoff between recall (which measures the ability to retrieve all relevant documents) and precision (which measures the ability to retrieve only documents which are relevant). The tradeoff between the two measures is a fundamental problem in Information Retrieval: as a system returns more documents, it finds more relevant ones and improves recall, but together with the relevant documents it retrieves more and more irrelevant ones and hurts precision. In the case of expert modeling, the profile set from a loose definition is larger but more ambiguous because many documents would be incorrectly associated because different people can have the same name. On the other hand, the profile set from a strict definition is smaller but more precise because retrieved documents are reliably associated with the person but at the same time valid documents are overlooked. Combining two expert definitions, *LAST* and *#uw(FIRST LAST)* gives better performance than either alternative separately (Table 4).

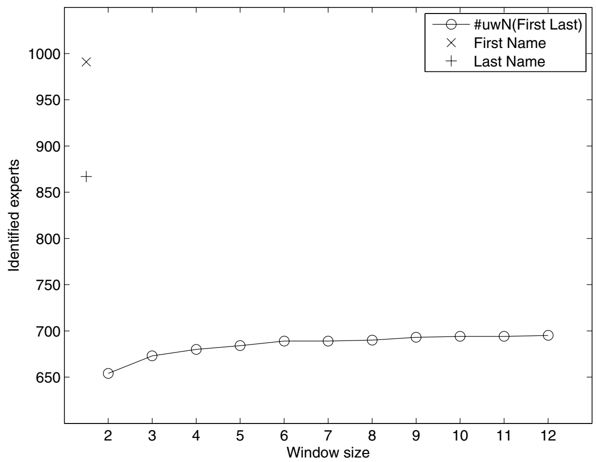

Figure 1. By relaxing the definition of an expert we find some information (at least one relevant document) about more experts.

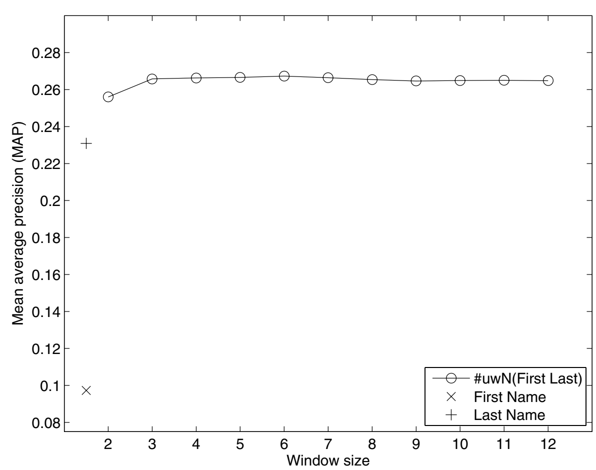

Figure 2. By relaxing the definition of an expert we incorrectly associate more documents with experts, resulting in a less precise model.

Expert Definition	Rel-ret	MAP	R-prec	RR1	P@10	P@20
`D0`	578	0.2443	0.2953	0.6300	0.3780	0.2990
`D1`	601	0.2786	0.3220	0.6458	0.4300	0.3350
`D2`	622	**0.2850**	0.3252	0.6496	0.4280	**0.3420**
`Best05`	571	0.2749	**0.3330**	**0.7268**	**0.4520**	0.3390

Table 4. Results of using different named entity definitions. We specify experts by their last name only (`D0`) and by both first and last name within text window of size 3 (`D1`), and we combine the two representations in (`D2`). The last row reports the best run in last year's TREC [8].

5 Conclusion and future work

We described a general entity modeling approach applied to finding people who are experts on a given topic. It is based on collecting evidence for expertise from multiple sources in a heterogeneous collection, using language modeling to find associations between documents and experts and estimate the degree of association, and finally integrating language models to construct rich and effective expert representations.

Our hierarchical approach combines the query-independent and query-dependent strategies to expert modeling to provide a greater flexibility in assembling information. Like a query-independent approach, it aggregates descriptions differently from different document formats but achieves this by combining probability distributions rather than concatenating text explicitly. Like a query-dependent approach, it preserves the information inherent in individual documents, such as structure and term proximity but considers only a subset of documents per expert rather than the entire collection.

Our approach provides a general framework for answering a variety of questions about experts, and we reported a series of experiments in which retrieval performance is incrementally improved. The results show that it can be successfully applied to search for experts in a multi-source repository.

Hierarchical language models can be used to describe entities other than people, for example places, organizations, events. Raghavan *et al.* [15] showed that automatically constructed probabilistic entity representations can be effective for a variety of tasks: fact-based question answering, classification into predefined categories, clustering and selecting keywords to describe the relationship between similar entities.

As future work, we plan to generalize the hierarchical expert models by modeling the relevance distribution differently for different experts. In our current work, we build a representation for each candidate expert based on a fixed number of associated documents. However, some people appear very frequently in the collection while others appear only in a few documents. The number of documents included in the profile set of an expert can be automatically adjusted to factor in this additional indicator of expertise.

6 Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval and in part by the Defense Advanced Research Projects Agency (DARPA), through the Department of the Interior, NBC, Acquisition Services Division, under contract number NBCHD030010. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

References

[1] The Lemur toolkit for language modeling and information retrieval. URL: `http://lemurproject.org/`.

[2] M. S. Ackerman. Augmenting organizational memory: a field study of answer garden. *ACM Transactions on Information Systems (TOIS)*, 16(3):203–224, 1998.

[3] K. Balog, L. Azzopardi, and M. de Rijke. Formal models for expert finding in enterprise corpora. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference*, 2006.

[4] K. Balog and M. de Rijke. Finding experts and their details in e-mail corpora. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 1035–1036, 2006.

[5] Y. Cao, J. Liu, S. Bao, and H. Li. Research on expert search at enterprise track of trec 2005. In *TREC-2005: Proceedings of the 14th Text REtrieval Conference*, 2005.

[6] N. Craswell, A. de Vries, and I. Soboroff. Overview of the trec 2005 enterprise track. In *TREC-2005: Proceedings of the 14th Text REtrieval Conference*, 2005.

[7] N. Craswell, D. Hawking, A.-M. Vercoustre, and P. Wilkins. P@noptic expert: Searching for experts not just for documents. In *Ausweb Poster Proceedings*, 2001.

[8] Y. Fu, W. Yu, Y. Li, Y. Liu, M. Zhang, and S. Ma. Thuir at trec 2005: Enterprise track. In *TREC-2005: Proceedings of the 14th Text REtrieval Conference*, 2005.

[9] H. Kautz, B. Selman, and A. Milewski. Agent amplified communication. In *AAAI-96: Proceedings of the 13th National Conference on Artificial Intelligence*, pages 3–9, 1996.

[10] V. Lavrenko and W. B. Croft. Relevance based language models. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference*, pages 120–127, 2001.

[11] D. Mattox, M. Maybury, and D. Morey. Enterprise expert and knowledge discovery. In *HCI '99: Proceedings of the 8th International Conference on Human-Computer Interaction*, pages 303–307, 1999.

[12] D. Metzler and W. B. Croft. A markov random field model for term dependencies. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference*, pages 472–479, 2005.

[13] R. Mukherjee and J. Mao. Enterprise search: Tough stuff. *Queue*, 2(2):36–46, 2004.

[14] P. Ogilvie and J. Callan. Combining document representations for known-item search. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference*, pages 143–150, 2003.

[15] H. Raghavan, J. Allan, and A. McCallum. An exploration of entity models, collective classification and relation description. In *LinkKDD '04: Proceedings of the 2nd International Workshop on Link Analysis and Group Detection in conjunction with the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004.

[16] Y.-W. Sim, R. Crowder, and G. Wills. Expert finding by capturing organisational knowledge from legacy documents. In *ICCCE '06: Proceedings of IEEE International Conference on Computer & Communication Engineering*, 2006.

[17] T. Strohman, D. Metzler, H. Turtle, and W. B. Croft. Indri: A language model-based search engine for complex queries. In *IA '05: Proceedings of the International Conference on Intelligence Analysis*, 2005.

[18] D. Yimam and A. Kobsa. Demoir: A hybrid architecture for expertise modeling and recommender systems. In *WETICE '00: Proceedings of the 9th International Workshop on Enabling Technologies*, pages 67–74, 2000.