Focus Article

Topic modeling for expert finding using latent Dirichlet allocation

Saeedeh Momtazi* and Felix Naumann

The task of expert finding is to rank the experts in the search space given a field of expertise as an input query. In this paper, we propose a topic modeling approach for this task. The proposed model uses latent Dirichlet allocation (LDA) to induce probabilistic topics. In the first step of our algorithm, the main topics of a document collection are extracted using LDA. The extracted topics present the connection between expert candidates and user queries. In the second step, the topics are used as a bridge to find the probability of selecting each candidate for a given query. The candidates are then ranked based on these probabilities. The experimental results on the Text REtrieval Conference (TREC) Enterprise track for 2005 and 2006 show that the proposed topic-based approach outperforms the state-of-the-art profile- and document-based models, which use information retrieval methods to rank experts. Moreover, we present the superiority of the proposed topic-based approach to the improved document-based expert finding systems, which consider additional information such as local context, candidate prior, and query expansion.

How to cite this article:
WIREs Data Mining Knowl Discov 2013, 3: 346–353 doi: 10.1002/widm.1102

INTRODUCTION

Finding people who are expert in a specific subject is a common task that can be used for many applications: Conference program committees are interested to automatically assign submitted papers to reviewers who are the best fit to each paper; employers in large companies look for experts in different fields to assign them the projects that are related to their fields; online knowledge sharing communities, specially question answering forums, need relevant expert users to answer questions[1]; students search for professors who are expert in the research areas they want to pursue their education. Having a system that can answer the following question accurately is an ultimate goal for addressing such information needs: "Who has knowledge about subject X?," where subject *X* is usually expressed as a (keyword) query.

There is a massive amount of data on the Web that provides useful information about people and can be used by a system to learn about their field of expertise: database records, e-mail exchanges, blogs, internal and external pages of organizations, meeting agendas, etc. In all of these documents, the experts' names and/or their e-mail addresses are presented in the same document that includes the subjects related to their work. Different techniques are proposed to use such information for finding experts in various domains. The available techniques focus on two approaches:

- *Profile-based approaches*, which build a profile for each candidate and then rank the candidates based on the relevance of their profile to the input query. In this model, the system first searches for candidates' names to build profiles using all documents that are related to each candidate and then searches for query terms.
- *Document-based approaches*, which retrieve all relevant documents to the input query

The authors have declared no conflicts of interest in relation to this article.

Present Address of S. Momtazi: Ubiquitous Knowledge Processing Lab (UKP-DIPF), German Institute for Educational Research and Educational Information, Frankfurt/Main, Germany.

*Correspondence to: momtazi@ukp.informatik.tu-darmstadt.de.

Hasso Plattner Institute, Potsdam University, Potsdam, Germany

DOI: 10.1002/widm.1102

and then find the association between the candidates and the retrieved documents. In this model, the system first searches for documents that contain query terms, and then looks for candidates' names that appear in these documents.

Based on the results reported by Balog et al.[2] on the test sets of the 2005 and 2006 edition of the Text REtrieval Conference (TREC) Enterprise track[3,4] the latter approach outperforms the former. One potential reason is that in the profile-based approach all documents that contain candidate's name are included in the candidate profile, even though they have different degrees of relevancy, which leads to introducing noise to the system with the documents that are not completely related to the target candidate. In the document-based approach, however, documents are used as a bridge between queries and candidates and their contributions are defined based on the number of terms they contain. Although the document-based approach results in reasonable performance for expert finding systems compared to the profile-based approach, it still suffers from the typical word mismatch problem. Because this model only works based on the co-occurrence of query words and candidates mentioned in documents; i.e., if a query word is related to other words in the vocabulary and those words are associated with a candidate, then the model is not able to capture the association between the query word and the candidate name. Such a problem indicates that documents alone are not enough to connect queries and candidates. Query expansion is one of the potential solutions to reduce this problem, but it also fails to discover *latent relations* between candidates and queries.

For this reason, a more sophisticated model is desirable to capture the relationship between candidates and users' queries rather than their co-occurrences in documents. To this aim, we build a more intelligent expert finder by exploring topics that are hidden in document collections and use the extracted topics instead of original documents to bridge candidates and queries.

Topic modeling using latent Dirichlet allocation (LDA)[5] is one of the most popular techniques that has been successfully applied in many text mining tasks. This model can capture the main topics of a document collection by (1) modeling each document as a probability distribution, which indicates the likelihood that it expresses each topic, and (2) presenting each topic as a set of words. Having both query terms and candidate names as vocabulary items, the latter aspect will be a great help for expert finding in such a way that the name of candidates who are expert in a field should be represented in the same topics as the query terms that ask about that field of expertise; i.e., if a candidate name and a query term appear in the same topic(s) with high probability, it is very likely that the candidate is an expert for the given query. In our proposed model, we first capture the main topics of a corpus using the LDA algorithm and then use the topics to calculate the relationship between each candidate and each field of expertise.

OVERVIEW OF EXPERT FINDING

Expert finding is one of the most active topics in the past years. It became more popular by defining as a sub-task in the Enterprise track of the Text REtrieval Conference TREC[a] in 2005. The task includes a list of expert candidates to be ranked based on a set of queries using a collection of documents.

The main goal of an expert finding system is determining how likely the candidate C is an expert given the input query Q. By calculating this probability for all candidates in the search space, the system can rank the candidates in a descending order of this probability. As a result, the main challenge in expert finding is to accurately estimate $P(C|Q)$. Using Bayes' theorem, this probability is formulated as follows:

$$P(C|Q) = \frac{P(Q|C) \cdot P(C)}{P(Q)} \tag{1}$$

where $P(Q|C)$ is the probability of generating query Q given the available information about candidate C. $P(C)$ is the prior probability of candidate C, and $P(Q)$ is the probability of query Q. Since $P(Q)$ is constant for a given query, it can be ignored and the formula will reduce to

$$P(C|Q) \propto P(Q|C) \cdot P(C) \tag{2}$$

According to this equation, the main factor for ranking candidates is the probability of generating the input query given each candidate. In addition, the prior belief on each candidate might also affect the ranking. Most research on expert finding focuses on the first factor and leaves the second factor as a uniform distribution over all candidates. There are, however, other works that show the positive effect of candidate prior on overall ranking.[6]

The majority of expert finding systems use either a profile-based approach or a document-based approach for calculating $P(Q|C)$. Profile-based approaches provide a profile for each candidate in the first step and then use these profiles to rank candidates based on the input query. In other words, in this approach, also called a query-independent model, a profile model is first created for each expert candidate *C* using documents that are related to the candidate. Then, the likelihood of the input query is estimated given the profile model. The P@noptic system[7] was one of the first systems that used this approach for ranking experts. Petkova and Croft[8] developed this approach as the baseline of their system and improved the method by hierarchical language models. Balog and de Rijke[9] proposed a profile-based model for expert finding using information retrieval. In their proposed model, a candidate's skill is represented as a score over documents that are relevant given a knowledge area. The relevance of a document is estimated using standard generative language model techniques (how likely a document would generate a certain knowledge area).

In the other approach, document-based expert finding, instead of creating a profile for each expert, the system uses documents to connect candidates to queries. In this approach, also known as a query-dependent model, all documents are first ranked based on their relevance to the input query. Then the association between candidates and ranked documents is estimated based on candidates' occurrence in documents. The developed system at MITRE[10] was the first attempt using this approach. Later on, Fang and Zhai[6] presented a general probabilistic model for expert finding and showed how the document-based model can be adapted in this schema. Balog et al.[2] applied this approach to a language model–based framework for expert finding. They also used the profile-based approach in their system and showed that the document-based approach performs better than the profile-based model. Serdyukov and Hiemestra[11] proposed a hybrid model for expert finding which combines both profile- and document-based approaches. Balog et al.[12] proposed another approach, called topic modeling, for expert finding. Their definition of topic, however, is different from ours. The term *topic* in their work refers to query words that users use to search for experts, whereas in the present work we use the term *topic* as a set of concepts that are extracted from a collection using a topic modeling algorithm. In the proposed approach by Balog et al., instead of modeling candidate profiles or documents, they built a model for each input query and used this model to calculate the probability of candidates given queries. Their approach is similar to the document likelihood method, which is used in language model–based information retrieval. Based on their results, this model underperforms the profile- and document-based approaches. The main reason of its poor performance is the sparsity of the models built from the queries.

In addition to the above methods that are based on information retrieval, there are other works on expert finding that use different techniques for ranking candidates. Macdonald and Ounis[13] proposed a data fusion model for expert finding. Rode et al.[14] proposed a graph-based method in which all documents, candidates, and their associations are represented in a graph, and candidates are ranked using their proposed relevant propagation methods. Li et al.[15] used textual information and social links to discover expertise network in online communities. They found expert users based on the number of documents posted or commented on a specific topic by the user and the number of times those documents are commented or retweeted, forwarded by other influential users. In addition, they analyzed the social links between experts using social network analysis to construct an expertise network on the specific topic.

LATENT DIRICHLET ALLOCATION

The idea of topic modeling in a collection of documents is to create a coarse-grained representation of documents and words. Extracting a set of topics from a corpus, each document is presented as a probabilistic mixture of topics. In addition, topics themselves are probabilistic distributions over words.

LDA was first introduced by Blei et al.[5] and has become one of the most popular techniques in topic modeling. This model has been widely used in many machine learning, natural language processing, and information retrieval applications. Wei and Croft[16] applied this model to language model–based information retrieval and compared it with probabilistic latent semantic indexing and cluster-based retrieval. Griffiths and Steyvers[17] used LDA for capturing scientific topics in a collection of documents. The model was used by Rosen-Zvi et al.[18] for extracting author-topic models. In their model in addition to the statistics of words within documents, the information about the authors of each document was used and each author was represented by a probability distribution over topics. For multiauthor papers, they considered a mixture of the distributions associated with the authors. The idea of author-topic modeling has been later utilized by Linstead et al.[19] to mine developers' contributions from a given subset of the Eclipse source code. Lin and He[20] applied topic modeling to sentiment analysis. Chrupała[21] utilized LDA as a word clustering technique for named entity recognition, morphological analysis, and semantic relation classification. LDA modeling was also applied

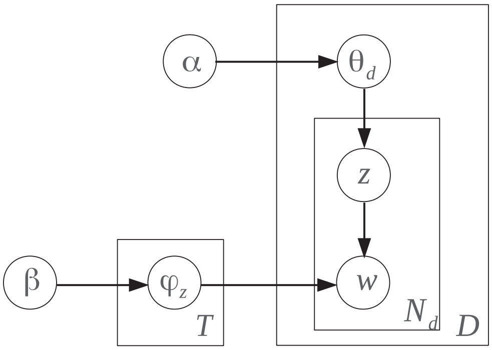

FIGURE 1 | Graphical representation of LDA topic models.

to expert finding in a metadata corpus by Wu et al.[22] and in R&D bibliographic data by Kongthon et al.[23] In their proposed models, however, topic modeling was only used in calculating the probability of query words given documents within the document-based framework and topic models are not considered as the core part of their system.

LDA is a generative, probabilistic hierarchical Bayesian model that induces topics from a document collection in three steps[5]:

1. Each document in the collection is distributed over topics that are sampled for that document based on a Dirichlet distribution.
2. Each word in the document is associated with one single topic based on this Dirichlet distribution.
3. Each topic is represented as a multinomial distribution over words that are assigned to the sampled topic.

The graphical model of LDA is shown in Figure 1. In LDA, ϕ is a matrix that presents the distribution of T topics over W vocabulary words from a Dirichlet prior with parameters β. And θ is a matrix that denotes the mixture distribution of D documents over these T topics from a Dirichlet prior parameterized by α. To generate each word token w in document d, a topic z is drawn from the topic distribution of the corresponding document θ_d, whereas the word w itself is drawn from the word distribution of the chosen topic ϕ_z. To extract topics using LDA, we need to estimate ϕ and θ, which provide information about the distribution of documents over topics and topics over words. Different algorithms have been proposed to estimate these parameters, such as expectation propagation, variational inference,[5] and Gibbs sampling.[17] Among these algorithms, Gibbs sampling has been shown as a simple, effective approach for this goal. We also use this model in our experiments.

EXPERT FINDING USING LDA TOPICS

As shown in previous research on expert finding, the profile-based approach underperforms the document-based approach, as a result of the noisy documents that are normally added to the candidate profile; i.e., in the profile-based approach, all documents that include candidate's name are normally considered as relevant documents and they are used to express a candidate's profile. The document-based approach, however, is more realistic and supports the connections between candidates and queries. This approach, which is based on the typical information retrieval techniques, also suffers from a simplification assumption: Connections between candidates and queries are based on their co-occurrence in documents. As a result, the model cannot support latent connections between candidates and queries, since it only relies on the query words for ranking experts and it does not consider any semantic concepts that are hidden in queries and documents.

This problem can be solved by extracting latent variables that correspond to a set of topics. We model this phenomenon with LDA topic modeling. In the topic-based approach, we use sets of words, namely topics, to bridge queries and candidates and avoid the disadvantage of considering direct co-occurrence of query words and candidate names, which is the main assumption in the document-based approach.

As mentioned, in LDA-based topic modeling each topic is a probabilistic distribution over words which can be used for capturing words relationship. In this model, it is assumed that the set of words which have the main contribution in representing a topic are conceptually related and they all are talking about the same concept. We benefit from this aspect of LDA to model the relationship between candidates and queries; i.e., instead of using documents directly to connect candidates and queries, we perform the expert finding process in two steps: (1) extracting a set of latent topics from the whole document collection and (2) using these topics to connect candidates and queries.

The process of topic extraction in the first step is based on the LDA algorithm, which is done as an off-line process. The document collection is only used in this step, and it is not required anymore in the next step. In the second step, the extracted topics are used to calculate the probability of query Q given candidate C. Thus, $P(Q|C)$ is calculated based on the topics that are distributed over query words and candidate names:

$$P(Q|C) = \sum_{t \in T} P(Q|t, C) \cdot P(t|C) \quad (3)$$

where $P(Q|t, C)$ is the probability of generating query Q given topic t and candidate C. By assuming conditional independence between Q and C, $P(Q|t, C)$ is simplified to $P(Q|t)$. Considering the input query as a bag-of-word model for calculating the probability of query Q given topic t, $P(Q|t)$ is calculated as follows:

$$P(Q|t) = \prod_{q \in Q} P(q|t)^{\#_q^Q} \quad (4)$$

where q is query term and $\#_q^Q$ is the number of times q occurs in Q.

By using Bayes' theorem for calculating the probability of topic t given candidate C, $P(t|C)$ is estimated as follows:

$$P(t|C) \propto P(C|t) \cdot P(t) \quad (5)$$

where $P(t)$ is the prior probability of topic t. Since in our model we do not make any difference between topics, this probability is considered uniform. Applying Eqs. (5) and (4) to Eq. (3), we have the following formula to estimate $P(Q|C)$:

$$P(Q|C) = \sum_{t \in T} \left[\prod_{q \in Q} P(q|t)^{\#_q^Q} \right] \cdot P(C|t) \cdot P(t) \quad (6)$$

Since the topic prior probability $P(t)$ is assumed as uniform, the main focus of this work is calculating the probability of query words given each topic $P(q|t)$ and the probability of candidates given each topic $P(C|t)$.

As described in the preceding section, by extracting topics from collection of documents using LDA, each topic can be modeled as a multinomial distribution over words. This distribution is estimated as ϕ in the LDA algorithm and shows the relation between words that are in the same topics. We use this distribution to calculate $P(q|t)$ and $P(C|t)$, since both query terms q and candidate names C are available in our vocabulary. As a result, $P(q|t)$ and $P(C|t)$ are estimated by ϕ_{qt} and ϕ_{Ct} respectively.

Like any other task that deals with statistical modeling, we have zero probabilities in this model. To avoid zero probabilities, we employ Jelinek–Mercer smoothing.[24] By using this smoothing method, we can interpolate a background probability with the original probability. As a result, even if the original probability is zero, the background probability keeps the overall estimation greater than zero. Applying Jelinek–Mercer smoothing to Eq. (6) results the following formula:

$$P(Q|C) = \sum_{t \in T} \left[\prod_{q \in Q} \{(1-\lambda) \cdot P(q|t) + \lambda \cdot P(q)\}^{\#_q^Q} \right] \cdot P(C|t) \cdot P(t) \quad (7)$$

where $P(q)$ is the background probability of the query word q estimated based on the frequency of the word in the corpus.[25]

Comparing our proposed model with topic-based information retrieval using LDA,[16] in the information retrieval task, the extracted topics are used to connect a query to a document. In this model, LDA is trained on an external corpus. Then the extracted topics are used for information retrieval; whereas the to-be-retrieved documents are used in the retrieval step, i.e., the distribution of topics over words (ϕ) is used for estimating $P(q|t)$ and the distribution of documents over topics (θ) is used for estimating $P(t|d)$. In our model, however, to-be-retrieved documents are not used in the retrieval step (as you can see, parameter d is not available in our formula). Instead, we only use these documents for training LDA, i.e., to-be-retrieved documents are used as a corpus to extract topics in an off-line process. Then, in the retrieval step, we only use the distribution of topics over words (ϕ) for estimating both $P(q|t)$ and $P(C|t)$ without requiring the distribution of documents over topics (θ). We believe that using to-be-retrieved documents for training LDA has no conflict with real assumptions, since the document models are not used in the retrieval step anymore and training LDA is the only place that we use the documents' contents. Comparing our model with the normal document-based model for expert finding, the LDA model helps us to find hidden concepts that connect queries to the candidates.

EVALUATION

Data Collection

To evaluate our proposed model, we used the test sets of the TREC Enterprise track. The Enterprise track at TREC ran the expert finding task in 2005 and 2006.[3,4] The 2005 and 2006 test sets contain 50 queries and 55 queries, respectively. The document collection used in both years to answer the queries is the W3C corpus.[b] The total number of documents in this corpus is 331,037, containing six different document types crawled from the W3C Web site: e-mail forum, code, Web, wiki, personal pages, and miscellaneous.

TABLE 1 | MAP of LDA-Based Topic Modeling Compared to Language Modeling[2] and Probabilistic Modeling[6]

Model	2005	2006
Topic based using latent Dirichlet allocation	0.248	0.471
Profile based using language model	0.188	0.321
Document based using language model	0.205	0.466
Document based using probabilistic model	0.172	0.204

Experimental Setup

As mentioned, in the proposed topic modeling, candidate names are also considered as vocabulary items. As a result, they can also appear in the list of words associated with each topic and their relation with query words can be discovered using the ϕ parameter. To this end, we need to match the names of people occurring in the documents with the list of candidates released for the expert finding task. Zhu[26] provided an annotation of candidate occurrences in the W3C document collection. This annotation considers various formats of candidate names/e-mails that are available in the corpus and tags each occurrence of a candidate by its corresponding ID. Following the other systems that we compared our model to, we used the same annotated corpus for our task. To this end, candidate IDs are used instead of their names.

For extracting topics from the corpus, we used the GibbsLDA++ toolkit,[c] a C/C++ implementation of LDA using Gibbs sampling. We used the default values of α and β in this toolkit; i.e., $50/T$ is used for α in which T is the number of topics, and 0.1 is used for β. For all experiments, the number of topics was set to 100, whereas varying this value and investigating the behavior of expert finding with different number of topics is left as future work. Since Gibbs sampling is an iterative process, we need to define the number of iterations to extract the final set of topics. The default value in this toolkit is 2000, and we also used this value. Since Gibbs sampling is a randomized algorithm to infer the posterior distribution, we used the samples from different Markov chains with different initialization and the average of the ones from several Markov chains was used to calculate the probabilities.

Results

The results of the proposed topic modeling approach on TREC 2005 and 2006 queries are presented in Table 1 in which mean average precision (MAP) serves as evaluation metric. Comparing the results of our model on TREC 2005 and 2006 queries, we can see a significant difference in the performance of our system on 2005 and 2006 queries. This behavior, however, is congruent with other expert finding systems[3,4] due to the different difficulty levels and assessments in 2005 and 2006 queries.

To compare our model with the state-of-the-art models, the results of Balog et al.[2] language models using either the profile- or document-based approach are presented in Table 1. In addition, we presented the results of Fang and Zhai[6] probabilistic models using the document-based approach. The results show that our topic modeling approach outperforms both document- and profile-based approaches using the state-of-the-art language models and probabilistic models.

As mentioned, capturing latent relations between query words and candidate names is the main advantage of our proposed model. As an example, the following queries are available in 2005 data set:

- Q06: *Mobile Web Initiative Workshop Program Committee*
- Q14: *Rules Workshop program committee*

The query words *workshop* and *committee* are mainly associated with a topic which consists of the following words and candidate names:

- Top words in the topic are *Team, review, Conference, organization, Submission, Group, Consortium, Member, meeting, Advisory, Committee, members, workshop, Members, groups, Workshop, Chair, conference, participants, organizations, member, papers, group*
- Top candidates in the topic are *candidate-0005, candidate-0177, candidate-0191*

Having such words in this topic as well as the relevant candidate names will help our model to find the correct candidates who are experts in the fields of the corresponding input queries.

As another step of the evaluation, we compared our model with the improved document-based approaches. To improve the document-based language modeling for expert finding, Balog et al.[2] proposed a proximity model. In their proximity model, instead of using the whole documents, a window of text surrounding query terms is used for connecting candidates and query terms. To improve the document-based probabilistic modeling, Fang and Zhai[6] proposed a candidate prior model. In their candidate prior model, instead of using a uniform priority for all candidates, candidates with high occurrence of their

TABLE 2 | MAP of LDA-Based Topic Modeling Compared to Improved Language Modeling[2] and Probabilistic Modeling[6]

Model	2005	2006
Topic based using latent Dirichlet allocation	0.248	0.471
Document based using language model + proximity model	0.219	0.454
Document based using probabilistic model + candidate prior	0.196	0.334
Document based using probabilistic model + candidate prior + expansion	0.204	0.359

e-mail address in documents receive a higher prior probability. Moreover, Fang and Zhai used a query expansion technique using model-based feedback to improve their system performance. We evaluated our topic-based approach against these improved models. The results of the comparison are presented in Table 2. As can be seen in the table, even though our topic-based approach uses neither the proximity model, the candidate prior, nor the query expansion, the model is still superior to the improved techniques. As mentioned, for the LDA algorithm, we used the whole documents to extract topics and did not consider local contexts to model proximity. We also did not consider any prior probability for candidates and assumed a uniform distribution over all candidates for calculating $P(C)$. All of these methods, however, can be applied to our topic-based approach as possible ways for further improvement of our system.

CONCLUSIONS

We proposed a novel approach for expert finding using topic modeling. Although most of the state-of-the-art methods directly use documents to connect candidates and query terms, we found that documents can be used indirectly to extract main topics that contribute to a corpus. The extracted topics are then used in our system to bridge candidates and query terms. The experimental results showed that the proposed model outperforms the state-of-the-art language modeling and probabilistic modeling, which both directly use documents for expert finding. Moreover, we showed the superiority of our topic-based approach to the improved methods, which use proximity model, candidate prior, or query expansion.

NOTES

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

[b] http://research.microsoft.com/en-us/um/people/nickcr/w3c-summary.html

[c] http://gibbslda.sourceforge.net

REFERENCES

1. Zhu H, Chen E, Cao H. Finding experts in tag based knowledge sharing communities. In: Xiong H, Lee W, eds. *Knowledge Science, Engineering and Management. Lecture Notes in Computer Science*, volume 7091. Berlin: Springer; 2011, 183–195.
2. Balog K, Azzopardi L, de Rijke M. A Language modeling framework for expert finding. *Inform Process Manag* 2008, 45:1–19.
3. Craswell N, Vries APD, Soboroff I. Overview of the TREC-2005 Enterprise Track. In: *Proceedings of the Text REtrieval Conference (TREC)*. Gaithersburg, MD; 2006.
4. Soboroff I, Vries APD, Craswell N. Overview of the TREC 2006 Enterprise Track. In: *Proceedings of the Text REtrieval Conference (TREC)*. Gaithersburg, MD; 2007.
5. Blei DM, Ng AY, Jordan MI, Lafferty J. Latent Dirichlet allocation. *J Mach Learn Res* 3, 2003.
6. Fang H, Zhai C. Probabilistic models for expert finding. In: *Proceedings of European Conference on Information Retrieval (ECIR)*. Berlin: Springer; 2007, 418–430.
7. Craswell N, Hawking D, Vercoustre A-M, Wilkins P. P@NOPTIC Expert: searching for experts not just for documents. In: *Proceedings of the Australian World Wide Web Conference (Ausweb)*. Coffs Harbour, NSW; 2001, 21–25.
8. Petkova D, Croft WB. Hierarchical language models for expert finding in Enterprise Corpora. In: *Proceedings of the IEEE International Conference on Tools With Artificial Intelligence (ICTAI)*. Piscataway, NJ: IEEE; 2006.
9. Balog K, de Rijke M. Determining expert profiles (with an application to expert finding). In: *Proceedings of the International Joint Conference on Artifical Intelligence (IJCAI)*. Hyderabad, India; 2007, 2657–2662.
10. Mattox D, Maybury M, Morey D. Enterprise expert and knowledge discovery. In: *Proceedings of the 8th International Conference on Human-Computer Interaction (HCI)*. Munich, Germany; 1999, 303–307.
11. Serdyukov P, Hiemstra D. Modeling documents as mixtures of persons for expert finding. In: *Proceedings of European Conference on Information Retrieval (ECIR)*. Berlin: Springer; 2008, 309–320.

12. Balog K, Bogers T, Azzopardi L, de Rijke M, van den Bosch A. Broad expertise retrieval in sparse data environments. In: *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York: ACM; 2007, 551–558.

13. Macdonald C, Ounis I. Voting techniques for expert search. *Knowl Inform Syst* 2008, 16:259–280.

14. Rode H, Serdyukov P, Hiemstra D, Zaragoza H. Entity ranking on graphs: studies on expert finding. *Technical Report TR-CTIT-07-81*, University of Twente, Enschede, the Netherlands, 2007.

15. Li Y, Ma S, Zhang Y, Huang R. Expertise network discovery via topic and link analysis in online communities. In: *IEEE 12th International Conference on Advanced Learning Technologies (ICALT), 2012;* Piscataway, NJ: IEEE; 2012, 311–315.

16. Wei X, Croft WB. LDA-based document models for ad-hoc retrieval. In: *Proceedings of the international ACM SIGIR Conference on Research and Development in Information Retrieval*. New York: ACM; 2006, 178–185.

17. Griffiths TL, Steyvers M. Finding scientific topics. *Proc Nat Acad Sci USA* 2004, 101:5228–5235.

18. Rosen-Zvi M, Chemudugunta C, Griffiths T, Smyth P, Steyvers M. Learning author-topic models from text corpora. *ACM Trans Inform Syst* 2010, 28.

19. Linstead E, Rigor P, Bajrachary S, Lopes C, Baldi P. Mining Eclipse developer contributions via author-topic models. In: *Fourth International Workshop on Mining Software Repositories (MSR)*. Minneapolis, MN; 2007, 30–33.

20. Lin C, He Y. Joint sentiment/topic model for sentiment analysis. In: *Proceedings of the ACM Conference on Information and Knowledge Management (CIKM)*. New York: ACM; 2009, 375–384.

21. Chrupała G. Efficient induction of probabilistic word classes with LDA. In: *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP)*. Chiang Mai, Thailand; 2011, 363–372.

22. Wu H, Pei Y, Yu J. Hidden topic analysis based formal framework for finding experts in metadata corpus. In: *Proceedings of the IEEE/ACIS International Conference on Computer and Information Science*. Washington, DC: IEEE Computer Society; 2009, 369–374.

23. Kongthon A, Haruechaiyasak C, Thaiprayoon S. Expert identification for multidisciplinary R&D project collaboration. In: *Portland International Conference on Management of Engineering and Technology (PICMET)*. Portland, OR; 2009, 1474–1480.

24. Jelinek F, Mercer R. Interpolated estimation of Markov source parameters from sparse data. In: *Proceedings of the International Workshop on Pattern Recognition in Practice*. Amsterdam, The Netherlands; 1980, 381–397.

25. Zhai C, Lafferty J. A study of smoothing methods for language models applied to information retrieval. *ACM Trans Inform Syst* 2004, 22(2): 179–214.

26. Zhu J. W3C corpus annotated with W3C people identity, 2006, http://ir.nist.gov/w3c/contrib/W3Ctagged.html. Accessed 2012.