Information Processing and Management xxx (2017) xxx-xxx

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com

On dynamicity of expert finding in community question answering

Mahmood Neshati*, [a], Zohreh Fallahnejad[b], Hamid Beigy[b]

[a] Faculty of Computer Science and Engineering, Shahid Beheshti University, G.C, Tehran, Iran
[b] Computer Engineering Department, Sharif University of Technology, Iran

A B S T R A C T

Community Question Answering is one of the valuable information resources which provide users with a platform to share their knowledge. Finding potential experts in CQA is beneficial to several problems like low participation rate of the users, long waiting time to receive answers and to the low quality of answers. Many research papers focused on retrieving the expert users of CQAs. Most of them are taking expertise into consideration at the query time and ignore the temporal aspects of the expert finding problem. However, considering the evolution of personal expertise over time can improve the quality of expert finding. In many applications, it is beneficial to find the potential experts in future. The proper identification of potential experts in CQA can improve their skills and the overall user participation and engagement. Considering dynamic aspects of the expert finding problem, we introduce the new problem of *Future Expert Finding* in this paper.Here, given the expertise evidence in current time, we aim to predict the best ranking of experts in future. We proposed a learning framework to predict such ranking on StackOverflow which is currently one of the most successful CQAs. We examine the impact of various features to predict the probability of becoming an expert user in future time. Specifically, we consider four feature groups; namely, *topic similarity, emerging topics, user behavior* and *topic transition*. The experimental results indicate the efficiency of the proposed models in comparison with several baseline models. Our experiments show that the performance of our proposed models can improve the MAP measure up to 39.7% in comparison with our best baseline method. Moreover, we found that among all of these feature groups, user behaviors have the most influence in the estimation of future expertise probability.

1. Introduction

Expert Finding is one of the challenging problems which attracted a lot of attention in Information Retrieval community in the past few years. The problem of expert finding concerns itself with identifying persons with relevant knowledge on a given topic and ranking them according to their expertise score. Several studies have been conducted to solve this problem in different areas. Most of the existing approaches for expert finding have been proposed to identify experts in the environments such as academic (Deng, Han, Lyu, & King, 2012; Deng, King, & Lyu, 2008; Hashemi, Neshati, & Beigy, 2013; Neshati, Hashemi, & Beigy, 2014), organizations (Balog, Azzopardi, & de Rijke, 2009; Karimzadehgan, White, & Richardson, 2009), forums (Xu & Ramanathan, 2016), microblogs and social network (Neshati, Asgari, Hiemstra, & Beigy, 2013; Neshati, Hiemstra, Asgari, & Beigy, 2014; Zhang, Tang, & Li, 2007) and more recently question answering communities (Pal, 2015; , Farzan, Konstan, & Kraut, 2011; Riahi, Zolaktaf, Shafiei, & Milios, 2012; van Dijk, Tsagkias, & de Rijke, 2015). In these approaches, associated documents, social interactions and the personal activities of each candidate are considered as expertise evidence. Identification of knowledgeable persons in a specific topic has a great importance in many applications such as assigning a paper to reviewers (Liang & de Rijke, 2016; Neshati, Beigy, & Hiemstra, 2012; Neshati, Beigy, & Hiemstra, 2014), finding the right supervisor in university domain (Alarfaj, Kruschwitz, Hunter, & Fox, 2012), finding expert users in question answering community (Riahi et al., 2012) and expert team formation (Kargar & An, 2011; Lappas, Liu, & Terzi, 2009).

* Corresponding author.
Email addresses: m_neshati@sbu.ac.ir, mahmood.neshati@gmail.com (M. Neshati); zfallahnejad@ce.sharif.edu (Z. Fallahnejad); beigy@sharif.edu (H. Beigy)

http://dx.doi.org/10.1016/j.ipm.2017.04.002
0306-4573/© 2016 Published by Elsevier Ltd.

Most of these approaches are taking expertise into consideration in a single snapshot of the environment (at the query time) and ignore the temporal aspects of the expert finding problem. However, people usually change their interests and expertise topics over time. Modeling the evolution of personal expertise over time not only is an important field of Information Retrieval (IR), but it also can improve the quality of expert finding (Balog, Fang, de Rijke, Serdyukov, & Si, 2012; Daud, Li, Zhou, & Muhammad, 2010; Fang & Godavarthy, 2014; Rybak, Balog, & Nørvåg, 2014). Various factors can affect the dynamics of personal expertise. The uptrend or downtrend popularity of a topic, the background of a person and his/her behavior on exploring new areas, and finally, the similarity of topics and the probability of transition between them are some few important factors in modeling expertise in a dynamic environment.

Community Question Answering (CQA) websites such as *StackOverflow*[1] provide users with a useful platform for information sharing. Users can post questions and answers, leave comments, and provide feedback on the quality of others' posts by voting, commenting and selecting the accepted answer to their questions. Successful CQA websites include those general ones such as *Yahoo! Answers*[2] and *Quora*[3], and those domain-specific ones like *StackOverflow* and *Mathematics Stack Exchange*[4]. Finding relevant experts on CQA for a given question/topic can enhance the quality of answers and accordingly improves the user experience and happiness (Riahi et al., 2012). On the other hand, one of the main problems of CQA services is the low participation rate of the users. Developing an expert finding system for intelligently routing newly posted questions can dramatically reduce the ratio of unanswered questions (Asaduzzaman, Mashiyat, Roy, & Schneider, 2013). More importantly, the job listings (SOFJobList, 2016) and CV Search (SOFCandidateSearch, 2016) (i.e. expert search) are two main revenue streams in StackOverflows business model which indicate the importance of an efficient expert finding system for such CQAs.

Finding experts in a CQA is a challenging task because of the following reasons:

- Only a small portion of users are responsible for answering a notable number of questions. It makes finding rising stars and potential experts quite difficult in CQAs (Daud, Ahmad, Malik, & Che, 2015).
- Emerging technologies like mobile programming and even small changes in the specification of some programming technologies can affect the behavior of CQAs' users (Linares-Vásquez, Bavota, Di Penta, Oliveto, & Poshyvanyk, 2014). It means that the capturing and modeling the dynamical aspects of expert finding in CQAs are more crucial in comparison with other expert finding environments like a bibliographic network in which people change their interests more slowly.
- The quality of user generated content in CQAs is not uniform for all users. Thus the quality of expert finding algorithms which depend on the quality of documents (i.e. questions and answers) may be indirectly affected. As a result, there are many research studies on detection and prediction of high-quality content on CQAs like StackOverflow (Ravi, Pang, Rastogi, & Kumar, 2014; Toba, Ming, Adriani, & Chua, 2014; Yao et al., 2015).

CQAs are dynamic environments because of the massive daily posts, joining new users, changing in their activities and interests, emerging new topics and the uptrend or downtrend of topics. For example, emerging technologies may make the existing ones obsolete; consequently, people change their skills and expertise. On the other hand, the success of CQA platforms is highly dependent on the users that can provide high-quality answers to the most difficult questions posted, however, this type of user is very rare (Riahi et al., 2012). As a result, the *prediction* and the nurture of users with topical expertise is becoming an important research topic in recent years (Pal, Farzan, Konstan, & Kraut, 2011; Pal, Harper, & Konstan, 2012a; van Dijk et al., 2015).

While these recent studies (Pal et al., 2011; Pal et al., 2012a; van Dijk et al., 2015) focus on the *early expertise detection* problem on CQA, we introduce the new problem of *Future Expert Finding* which focuses on the ranking of potential topical experts in the future time given expertise evidence in the query time. Most of the expert finding approaches are taking expertise evidence (e.g. experts' documents, experts' actions, etc.) into consideration in a single snapshot of the environment (i.e. at the query time) and neglect the temporal aspects of the expert finding problem. Therefore, the most immediate problem is how to model the dynamic and temporal aspects of the expert finding problem. Modeling the temporal characteristics of CQAs, not only helps us to analyze the changes in an expert's interests and expertise over the life cycle of users but more importantly, it allows us to predict their future expertise and accordingly improve the quality of CQA by routing the questions to potential future experts.

Considering dynamical aspects of the expert finding problem, in this paper, we propose two learning algorithms to predict the best ranking of experts in future time. We examine the impact of various features to predict the probability of becoming an expert user in future time. Specifically, we consider four feature groups; namely, *topic similarity, emerging topics, user behavior* and *topic transition* features. We found that all of these feature groups can be beneficial to predict the probability of becoming an expert. In order to test the proposed learning models, we use an automatically generated test collection from StackOverflow data collection. We compare our proposed models with the several baseline algorithms to predict the future experts. The results confirm the effectiveness of the proposed models. The main contributions of our paper are as follows:

[1] stackoverflow.com
[2] answers.yahoo.com
[3] www.quora.com
[4] math.stackexchange.com

- Introduction of the new problem of *Future Expert Finding in CQA* which focuses on the ranking of potential topical experts in the future time given the expertise evidence in query time.
- Proposition of two learning to rank approaches to solve the problem efficiently and effectively.
- Comparison of the proposed approaches with several baseline approaches on a real dataset.

The rest of this paper is organized as follows. Section 2 introduces the problem of future expertise finding. Section 3 reviews some prior research related to our work. Section 4 is devoted to a description of the background and preliminaries. Section 5 discusses the proposed approach for the future expert finding problem. Experimental results are presented in Section 6. We conclude this paper and point out future works in Section 7.

2. Future expert finding problem

The classical expert finding problem is about finding the ranking of people who are knowledgeable in a given topic. Successful probabilistic ranking models estimate the expertise probability of $p(e|q)$ for each candidate e for the given query q. The implicit assumption in this problem is that the expertise evidence is captured and analyzed at the query time. However, in many applications, it is beneficial to find the potential experts in future time. For example, the proper identification of potential experts in CQA could improve their skills and improve the overall quality of their participation in the community (Pal et al., 2011). In order to fill this gap, we focus on *Future Expert Finding Problem* which can be defined as follows:

For a given query q and a set of observed expertise evidence at time t_1, *rank expert candidates according to their knowledge about q at time* t_2, *while* $t_1 \leq t_2$.

Specifically, we want to estimate the probability of $p(e|q, CT = t_1, FT = t_2)$ to rank candidate e in future time ($FT = t_2$) while the expertise evidences are given at time $CT = t_1$. Future expert finding problem concerns with the ranking of people according to their potential expertise in future.

Several factors can affect the potential expertise of a candidate in future. Consider Fig. 1 that indicates the popularity of three topics over years on a subset of StackOverflow data collection. According to this Figure, the popularity of topics can be increasing (e.g. android), decreasing (e.g. java basics) or constant (e.g. logging) over a period of time[5]. Intuitively, we expect that the number of experts will be in an uptrend for emerging topics and a downtrend for disappearing topics.

Fig. 2 indicates the topic transition pattern between *android* topic and other topics. The vertical axis indicates the average number of incoming and outgoing experts[6] between *android* topic and other topics in a year. Three important results can be inferred from this Figure. First, there are only few related topics (i.e. *GUI* and *JSON*) which have high transition rate with *android* topic. This means that experts usually change their expertise area smoothly (i.e. they move to some related topics and avoid jumping to a completely new topic). Second, it is quite likely that people who are working on a topic (e.g. *android*) stay on the same topic over time. In other words, some conservative people usually stick towards the topics they prefer and do not change their expertise topics frequently. Finally, the number of incoming and outgoing experts between *android* and a given topic are almost the same which means that transition pattern can be described by the semantic similarity of topics.

We encode the above mentioned temporal aspects of CQA into a ranking model to solve the future expert finding problem.

3. Related work

In the past few years, expert finding problem attracted a lot of attention in the IR community. The expert finding problem has been studied in many environments such as organizations (Balog et al., 2009), universities (Balog, Bogers, Azzopardi, de Rijke, & van den Bosch, 2007), bibliographic networks (Hashemi et al., 2013; Neshati Neshati, Beigy, & Hiemstra, 2014), social networks (Neshati et al., 2013; Neshati, Hiemstra, Asgari, & Beigy, 2014), Wikipedia (Ziaimatin, Groza, Bordea, Buitelaar, & Hunter, 2014), LinkedIn Budalakoti and Bekkerman (2012), CQAs and even Instagram Pal, Herdagdelen, Chatterji, Taank, and Chakrabarti (2016). The contextual, behavioral and social information have been used in order to estimate the expertise score of each candidate.

Several researchers have addressed the problem of expert finding in CQAs. These researchers mainly focused on finding a group of experts to route newly asked questions with the objective of providing users with high-quality answers within a reasonable time. Riahi et al. (2012) proposed a framework for automatically routing a newly posted question to an expert user. They used statistical topic models to detect expert users in CQAs. Zhou, Cong, Cui, Jensen, and Yao (2009) utilized both the content and structures of the forum system for the efficient routing of a given question to the top-*k* potential experts in the forum. Li, Jin, and Shudong (2015) proposed a tag-LDA model to determine the user topic distribution and predicts the topic distribution of new questions. They considered user post contents, answer votes, ratio of best answers, and user relations to find an appro-

[5] Topics are inferred using LDA (Blei, Ng, & Jordan, 2003) and manually labeled.

[6] By expert on topic *m*, we mean candidates who answered more than 10 questions related to topic m which have been approved as the accepted answer by the questioner.

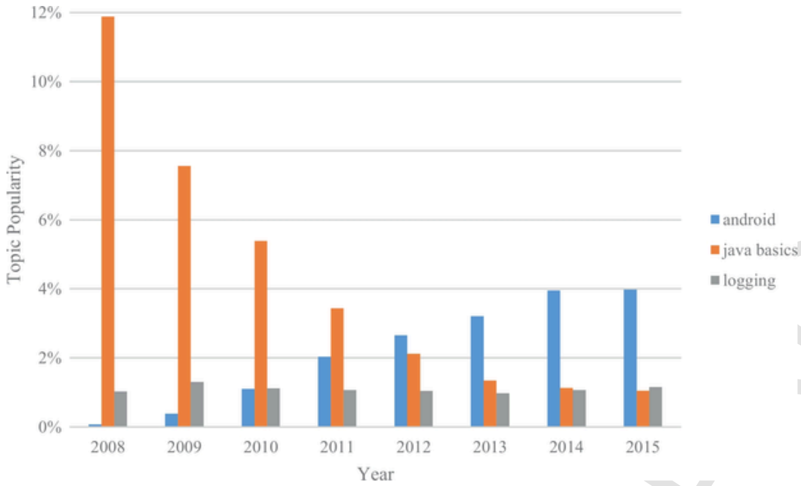

Fig. 1. Topic popularity in different years.

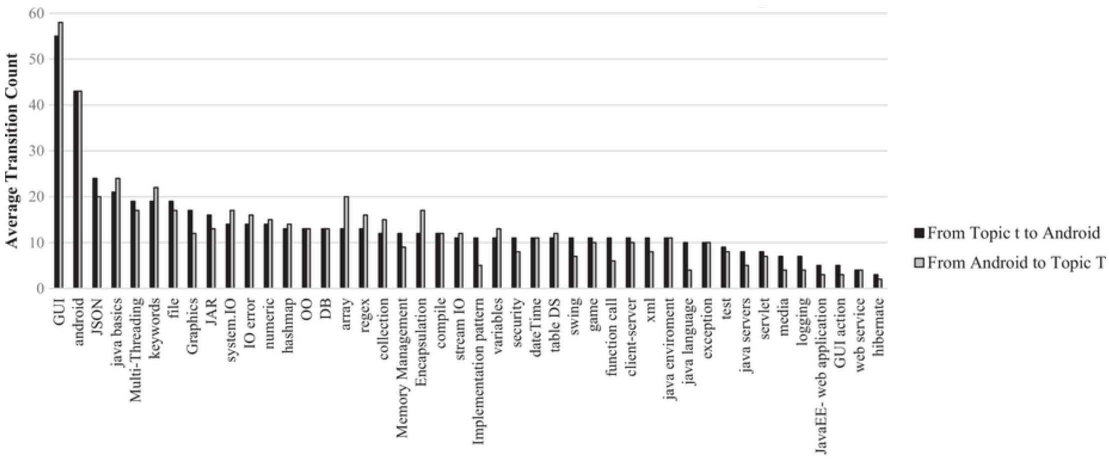

Fig. 2. Average number of incoming and outgoing experts between *android* topic and other topics.

priate user to answer a new question. In order to recommend new questions to a wider part of a community like newcomers or lurkers, Srba, Grznar, and Bielikova (2015) proposed a question routing method which analyses users non-QA data from CQA and external services (such as blogs, microblogs or social networking sites) as a supplement to QA activities in estimation of users expertise. Inspired by previous studies, Pal (2015) recently proposed a framework for finding relevant communities for a question by considering the problem of routing a question to the right community. The above mentioned approaches mainly focused on finding experts at the query time and ignored the dynamic aspects of expert finding.

One of the main aspects of the expert finding problem is expert identification. The primary question in the expert identification problem is to study the differences between users and to define characterization metrics of experts. Therefore, it is necessary to study behavioral patterns of users. Based on the different behavior pattern of experts, Yang, Tao, Bozzon, and Houben (2014) proposed a new metric for expert identification in the context of the StackOverflow. Considering the quality of the user contributions, they identified two class of users, namely most active users (the sparrows) and most knowledgeable users (the owls). Pal, Chang, and Konstan (2012) presented a temporal study of experts in CQA and analyzed the changes in their behavioral patterns over time. They explore the different evolution patterns exhibited by expert and ordinary users. For example, they showed that as the probability of providing the best answer increases for experts, while it decreases for ordinary users over time. By using unsupervised machine learning methods, they used these evolution patterns to distinguish experts from one another. Rowe (2013a) studied the changes of user activity based on their social and lexical properties through their lifecycles in the context of online community platforms. Patil and Lee (2016) studied users on Quora and identified three type of expert users based on the weekly changes in the number of answers they provide. They also used temporal features including daily changes in the number of followees, followers, edits, questions, and answers to improve the precision of expert detection. Furtado, Andrade, Oliveira, Brasileiro, 2013) analyzed user behaviors in five StackExchange Q&A websites and identified ten contributor profiles by using clustering methods. They studied these profiles to discover the behavioral changes of contributors over time. Studying user behavior can be beneficial to detect churners in CQAs. Rowe (2013b) proposed an approach to mine the lifecycle trajectories of users as a means to characterize user development along the various properties (in-degree, out-degree, posted terms), and demonstrate the utility of such trajectories in predicting churners. Rowe (2016) analyzed social and lexical user development based their prior behavior and the community in which they have interacted and developed a detection model to identify churners in online community platforms. While these studies focused on identification of the characteristics of experts, they can not be used directly for the problem of expert finding.

Studying the temporal aspects of user activities is not limited to CQAs. It had been used in product recommendation systems. Considering the evolution of user expertise and differences between novices and experienced users, McAuley and Leskovec (2013) proposed a latent factor recommendation system. Mukherjee, Lamba, and Weikum (2015) used the coupling between user experience, interest in specific item facets, writing style, and rating behavior to capture the users temporal evolution and proposed an individual recommendation approach which takes into account the users maturity level.

Identification of potential experts is the most related line of research to our work. In order to increase the quality of CQA services, some recent researches have focused on the detection of potential experts in the early stage of their lifecycle. Identifying these experts during the first few weeks of their joining help community managers to nurture and retain these potential users in the community. van Dijk et al. (2015) proposed a semi-supervised machine learning approach which uses textual, behavioral and time-aware features to measure whether a user shows signs of early expertise for a given topic. Pal et al. (2011) analyzed behavioral characteristics of newly joined users and used predictive and ranking algorithms to estimate their motivation and ability to help others. Pal, Harper, and Konstan (2012b) used selection preferences of users in the identification of community experts and potential experts. Sung, Lee, and Lee (2013) use user's expertise and availability with the notion of the answer affordance to measure the likelihood of becoming a contributive user. They utilized a user's productive vocabulary to mitigate the lack of available information since the vocabulary is the most fundamental element that reveals his/her knowledge.

Creating expertise profile is another approach to detect changes in personal expertise over the time. Rybak et al. (2014) introduced the concept of a hierarchical expertise profile as a weighted tree. They defined temporal expertise profile as a series of time-stamped hierarchical profiles and compared them to characterize important changes. Ziaimatin et al. (2014) proposed a domain-agnostic methodology for creating short-term and long-term profiles, while capturing the temporality in expertise. It should be mentioned that these approaches are only capable of detecting changes in expertise profile of each person not to predict their futures. While the expert profiling is a related problem to expert finding, some previous researches Balog and De Rijke (2007) indicate that the solution for expert profiling can not be directly used for the expert finding problem.

Daud, Li, Zhou, and Muhammad (2009) proposed a time topic modeling approach called Temporal-Author-Topic(TAT) whose objective is to formalize dynamicity of researchers interests over time. TAT can simultaneously model text, researchers and, time of research papers without changing the meaning of topics for different years, unlike Author-Topic(AT) model which does not incorporate time information and should be applied to each year separately. This approach is used to discover topically related researchers for different time periods. The modeling and formulation proposed in this study can not easily extend to solve the future expert finding problem.

As another related work, Yeniterzi and Callan (2015) proposed adapting temporal discounting models to expertise estimation methods for question routing. Two widely used expert finding approaches, Answer Count and Z_{Score}, were modified to use the available temporal information. They used available temporal information in CQA sites to make these existing approaches more effective for the task of question routing.

The most related research to our work is the study proposed by Fang and Godavarthy (2014) which will be described later. They proposed a probabilistic approach to model the temporal profile of candidates in a bibliographic network. Our work differs from this work in three ways: first, their method is basically an expert profiling and as mentioned before cannot be used directly for expert finding problem, second, they investigated their method on bibliographic network which is not comparable with StackOverflow in terms of both the size of data and the complexity of temporal aspects, and finally, they only evaluate the quality of their predictive language model does not provide expert finding or expert profiling evaluation.

4. Background and preliminaries

In this Section, we describe several methods of expert finding which can be used as baseline models to solve the future expert finding problem. The first method is the famous document-centric model proposed by Balog et al. (2009). This method completely ignores the dynamic aspects of the expert finding problem, but because of its satisfactory results and its solid mathematical foundation, we select it as one of the baseline models. The second method is the one proposed by Momtazi and Naumann (2013) which uses topic modeling to improve the quality of expert finding. This method also ignores the dynamic aspects of expert finding, but we select it as a baseline because the topic modeling approach is explicitly used in the third baseline as well as in our proposed model. The third baseline is the probabilistic dynamic expert profiling approach proposed by Fang and Godavarthy (2014) which models the dynamics of the personal expertise.

4.1. Document based model (DBM)

One of the most efficient models for topical expert finding is the document-centric model (also known as the *Model 2*) proposed by Balog et al. (2009). In this model, for a given query q, the relevance probability of an expert is determined by the following equation:

$$p(e|q) = \frac{p(e)}{p(q)} p(q|e),$$

where $p(e)$ is the prior relevance probability of expert candidate e, $p(q)$ is the occurrence probability of a query q and $p(q|e)$ represents the generation probability of query q given the expert candidate e. Since $p(q)$ is a constant value for all the candidates, it can be removed from the equation and we obtain:

$$p(e|q) \propto p(e) p(q|e).$$

The probability of query q given the expert candidate e, i.e. $p(q|e)$, can be estimated by taking the sum over all documents d associated with the candidate e:

$$p(q|e) = \sum_{d \in D_e} p(d|e) p(q|d,e),$$

where D_e indicates the set of documents associated with candidate e, $p(d|e)$ is the association strength of document d and candidate e. By assuming the conditional independence between candidate e and query q for a given document d, we can simplify this equation as follow:

$$p(q|e) = \sum_{d \in D_e} p(d|e) p(q|d).$$

Using Bayes' theorem applied to probability $p(d|e)$ and assuming uniform value for the prior probability of documents (i.e. $p(d)$ is equal for all documents), the final ranking probability can be written as follows:

$$p(e|q) \approx \sum_{d \in D_e} p(e|d) p(q|d).$$

We can use the described model to approximately solve the future expert finding problem introduced in Section 2. The extended model can rank potential experts in a future time t_2 given expertise events in time t_1 (i.e. query time), while $t_1 \le t_2$. This model ignores the dynamical aspects of the problem and simply use the current ranking of experts as their future ranking which can be a reasonable assumption in short time intervals. So, the probability of a candidate e being an expert in the future time $FT = t_2$ given the query q at the current time $CT = t_1$ is estimated as:

$$p\left(e|q, CT = t_1, FT = t_2\right) \approx p\left(e|q, CT = t_1, FT = t_1\right) \approx \sum_{d \in D_{e,t_1}} p(e|d) p(q|d), \tag{1}$$

where D_{e,t_1} is a subset of documents associated with candidate e observed at time t_1 and $p(e|d)$ is the association strength of document d and expert candidate e. The probability $p(e|d)$ equals to 1 if candidate e is the author of document d otherwise it will be zero. Similar to the original document-centric model proposed in Balog et al. (2009), the probability of $p(q|d)$ is estimated by the generation probability of q by the language model of document d. In our experiments, we use two definition for D_{e,t_1} as follows:

$$D1_{e,t_1} = \left\{d \in D \mid p(e|d) = 1 \text{ and } Time(d) \le t_1\right\} \tag{2}$$

$$D2_{e,t_1,a} = \left\{d \in D \mid p(e|d) = 1 \text{ and } t_1 - a \le Time(d) \le t_1\right\} \tag{3}$$

In above equations, *Time*(d) indicates the punishment time of document d and a is the length of time window prior the current time t_1. The difference between $D1_{e,t_1}$ and $D2_{e,t_1,a}$ is that in Eq. (3), only a subset of documents associated with e which is written in time interval $\left[t_1 - a, t_1\right]$ is considered as expertise evidences of candidate e while in Eq. (2) all associated documents which published prior t_1 are considered as the expertise evidence.

To sum up, we use two baseline algorithms based on the document-centric approach to solve the future expert finding problem as follows:

$$p_{B_1,ALL}\left(e|q,CT=t_1,FT=t_2\right)\approx\sum_{d\in D1_{e,t_1}}\prod_{w\in q}p\left(w|\theta_d\right)^{n(t,q)} \tag{4}$$

$$p_{B_1,RECENT}\left(e|q,CT=t_1,FT=t_2\right)\approx\sum_{d\in D2_{e,t_1,a}}\prod_{w\in q}p\left(w|\theta_d\right)^{n(t,q)}, \tag{5}$$

in which w indicates the query words and θ_d is the language model of document d. We use the JM-smoothing Zhai and Lafferty (2004), so the probability $p(w|\theta_d)$ is estimated as $p\left(w|\theta_d\right)=(1-\lambda)\,p\,(w|d)+\lambda p\,(w)$, where $p(w|d)$ is the maximum likelihood estimation of the occurrence of term w in document d, and $p(w)$ is the occurrence probability of term w in the document repository.

We refer to this method as Document Based Model (DBM) in the rest of this paper.

4.2. Topic based model

Following the idea of Momtazi and Naumann (2013), we can use topic modeling to improve the quality of future expert ranking problem. The model proposed in Momtazi and Naumann (2013) uses latent Dirichlet allocation (LDA) (Blei et al., 2003) to induce probabilistic topics from documents. In the first step, LDA method has been used to extract topics of each document. The extracted topics show 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 model proposed in Momtazi and Naumann (2013) can easily be extended as a baseline for future expert finding problem as indicated in the following equation:

$$p\left(e|q,CT=t_1,FT=t_2\right)\approx p\left(e|q,CT=t_1,FT=t_1\right)\approx\sum_{m\in \text{Topic}_{e,t_1}}p\,(e|m)\,p\,(q|m), \tag{6}$$

where Topic_{e,t_1} indicates the set of associated topics to candidate e at time t_1. Eq. (6) is similar to Eq. (1) with a difference that the retrieval units in Eq. (1) are documents but in Eq. (6) the retrieval units are topics. In Eq. (6), $p(e|m)$ indicates the association strength between candidate e and topic m which is estimated as follows:

$$p\,(e|m)=\frac{\text{the number of documents associated with candidate e and related to topic t}}{\text{the number of documents related to topic t}}.$$

In the above equation, we use a binary decision to count the number of related documents. Specifically, if a document has a non-zero weight for a given topic, we count it as an associated document to that specific topic.

Similar to Momtazi and Naumann (2013), in order to estimate $p(q|m)$, we use independence assumption of the query term q as follow:

$$p\,(q|m)=\prod_{w\in q}p(w|m)^{n(w,q)},$$

where $p(w|m)$ is the estimation of the occurrence of term w in topic m and $n(w, q)$ is the frequency of term w in query q.

We refer to this method as Topic Based Model (TBM) in the rest of this paper.

4.3. Temporal profile based model (TPBM)

In this Section, we described the probabilistic dynamic expertise profiling approach proposed by Fang and Godavarthy (2014) to solve the future expert finding problem. Similar to the baseline model proposed in Section 4.2, the retrieval units in this method are topics. The main difference between the method proposed in Fang and Godavarthy (2014) and the method proposed in Section 4.2 is that method (Fang & Godavarthy, 2014) probabilistically estimates the association between a candidate e and a topic m in a future time t_2; while the method described in Section 4.2 estimates the mentioned probability according to their association in current time t_1.

In this model, the probability of the future expertise of candidate e for the given query q is estimated as follows:

$$p\left(e|q,CT=t_1,FT=t_2\right)=\sum_{m\in Topic}p\left(q|m,CT=t_1,FT=t_2\right)p\left(m|e,CT=t_1,FT=t_2\right). \tag{7}$$

The probability $p\left(m \middle| e, CT=t_1, FT=t_2\right)$ in Eq. (7) indicates the probability of association between a candidate e and a topic m in future time t_2 while related evidences are observed in time t_1.

Assuming the Markov property, the mentioned probability is estimated by the following equation:

$$p\left(m \middle| e, CT=t_1, FT=t_2\right) = \sum_{n \in Topic} p_{t_1 \to t_2}(m|n, e)\, p_{t_1}(n|e), \tag{8}$$

where $p_{t_1}(n|e)$ is the probability of association between a topic n and a candidate e at time t_1 and $p_{t_1 \to t_2}(m|n, e)$ is the probability of topic transition of a candidate e from a topic n at time t_1 to topic m at time t_2.

According to the Markov assumption, in above equation, the association between topic m and candidate e in future time t_2 is only dependent on the expertise topics of e at time t_1 and is independent from his/her expertise topics prior to t_1.

In above equation the probability $p_{t_1}(n|e)$ indicates the association probability of topic n and candidate e in current time t_1, which according to Fang and Godavarthy (2014) estimated as the ratio of the number of times the topic n is associated with e divided by number of all topics associated with e.

The probability of $p_{t_1 \to t_2}(m|n, e)$ indicates the probability of topic transition of candidate e from topic n to topic m in time interval $[t_1, t_2]$. In the model proposed in Fang and Godavarthy (2014), in order to study the above transition probability, three factors are taken into consideration: 1) the personality of the expert in exploring new areas; 2) the similarity between the new area and the expert's current areas; 3) the popularity of the new area. They used the mentioned factors to estimate the transition probability between current and future topics.

We refer to this method as Temporal Profile Based Model (TPBM) in the rest of this paper.

5. The proposed models

As mentioned before, in order to solve the future expert finding problem, we should estimate the probability of $p\left(e \middle| q, CT=t_1, FT=t_2\right)$ to rank people at the future time t_2, given the query q and expertise evidences at the current time t_1.

According to Bayes' rule, we have:

$$p\left(e \middle| q, CT=t_1, FT=t_2\right) = \frac{p\left(e \middle| CT=t_1, FT=t_2\right)}{p\left(q \middle| CT=t_1, FT=t_2\right)} p\left(q \middle| e, CT=t_1, FT=t_2\right).$$

Assuming a constant value for $p\left(e \middle| CT=t_1, FT=t_2\right)$ and $p\left(q \middle| CT=t_1, FT=t_2\right)$, we can use $p\left(q \middle| e, CT=t_1, FT=t_2\right)$ as the future expertise score of candidate e.

Following the idea of topic based retrieval model (refer to Section 4.2), the probability of $p\left(q \middle| e, CT=t_1, FT=t_2\right)$ can be estimated using the following equation:

$$p\left(q \middle| e, CT=t_1, FT=t_2\right) = \sum_{m \in Topic} p\left(m \middle| e, CT=t_1, FT=t_2\right) p\left(q \middle| m, e, CT=t_1, FT=t_2\right),$$

in which $p\left(m \middle| e, CT=t_1, FT=t_2\right)$ indicates the expertise probability of candidate e on topic m in future time t_2 while we observe the associated documents of candidate e at time t_1 and $p\left(q \middle| m, e, CT=t_1, FT=t_2\right)$ indicates the probability of generation of query q in future time t_2 given topic m and candidate e. Similar to the idea of Momtazi and Naumann (2013), we assume the independence of query from the candidate given the topic (i.e. $p\left(q \middle| m, e, CT=t_1, FT=t_2\right) \approx p\left(q \middle| m, CT=t_1, FT=t_2\right)$). In order to simplify the computations, we follow (Fang & Godavarthy, 2014) and assume that the vocabulary of a topic remains fixed over time. Specifically, we can estimate probability $p\left(q \middle| m, e, CT=t_1, FT=t_2\right)$ as follows:

$$p\left(q \middle| m, e, CT=t_1, FT=t_2\right) \approx p(q|m) \approx \prod_{w \in q} p(w|m)^{n(w,q)}.$$

In order to estimate the probability of $p\left(m \middle| e, CT=t_1, FT=t_2\right)$, we follow the Markov assumption proposed by Fang and Godavarthy (2014) and assume that the topical expertise of candidate e in time t_2 is independent from his/her expertise at any time prior t_1. Hence, $p\left(m \middle| e, CT=t_1, FT=t_2\right)$ becomes:

$$p\left(m \middle| e, CT=t_1, FT=t_2\right) = \sum_{n \in Topic} p_{t_1 \to t_2}(m|n, e)\, p_{t_1}(n|e), \tag{9}$$

in which $p_{t_1}(n|e)$ indicates the topical expertise of candidate e at current time t_1 which can be estimated using Eq. (8) given in Section 4.3.

The critical part of the proposed model is the precise estimation of the $p_{t_1 \to t_2}(m|n,e)$ probability. This probability indicates the probability of transition of candidate *e* from topic *n* at time t_1 to topic *m* at time t_2. The predicted value for the topic transition probability can be plugged into the Eq. (9) to estimate the relevancy of each candidate on a topic in future.

We propose two main approaches to construct the ranking models, such that the model can sort future topics for each candidate according to several feature groups. In these models, the main goal is to precisely estimate probability $p_{t_1 \to t_2}(m|n,e)$.

In our first attempt, following the idea of point-wise learning Liu (2009), we cast the estimation of the transition probability into a classification problem that treats the relevant future topics of a given candidate as the positive data (i.e. candidates who actually transit to topic *m* at time t_2), and other topics as the negative data. More precisely, in this approach, each quintuple (e, m, n, t_1, t_2) is represented by a feature vector and the algorithm assigns a score to each vector instance proportional to $p_{t_1 \to t_2}(m|n,e)$
.

In our second approach, we follow the idea of pair-wise learning Liu (2009) to estimate the topic transition probability. In this approach, for each pair of future topics associated with a specific candidate, we determine the preference function of ranking (i.e. which topic should be scored higher) according to our data collection. Using these preferences, we train a ranking model; then for a given candidate, we can rank topics in future according to the topic transition probability. In the rest of this Section, we first explain the point-wise learning model and then we introduce the pair-wise learning model. At the end of this Section, we explain the feature vectors we used for training these two models.

5.1. Point-wise estimation

In this Section, we propose a discriminative method to predict the topic transition probability $p_{t_1 \to t_2}(m|n,e)$.

In our proposed model, we use a topic transition variable $c \in \{0, 1\}$ to denote how much the candidate *e* has chance to transit from the old topic *n* to new topic *m* in time interval $[t_1, t_2]$. Specifically, the probability of $p_\theta\left(c = 1|e, m, n, t_1, t_2\right)$ can be used as the topic transition estimator where θ is the unknown parameters that should be learned using training data.

In order to train our model, we generate a training set using the given data collection as follows:

$$TS_1 = \left\{\left(e, n, m, t_1, t_2, l\right) \mid l \in \{0, 1\}\right\}$$

In this training set, label *l* is equal to one if and only if the candidate *e* is an expert on topic *n* at time t_1 and an expert on topic *m* at time t_2 otherwise it will be zero.

In order to label candidates in our data collection as expert or non-expert, we follow the method proposed by van Dijk et al. (2015). By expert on topic *m*, we mean candidates who answered more than 10 questions related to topic *m* which have been approved as the accepted answer by the original questioner. The accepted answer on StackOverflow is illustrated by the green check mark next to the answer.

In order to generate training instances, for given t_1 and t_2, we partitioned candidates into four sets.

- S_1: This set includes candidates who are not expert on any topic at t_1 and t_2.
- S_2: This set includes candidates who are expert on at least one topic *n* at time t_1 but are not expert on any topic at time t_2.
- S_3: This set includes candidates who are expert on at least one topic *m* at time t_2 but are not expert on any topic at time t_1.
- S_4: This set includes candidates who are expert on at least one topic *n* at time t_1 and at least one topic *m* at time t_2.

In order to generate positive instances, we randomly select candidates from set S_4 and a pair of old-new topic as a positive instance and to generate negative instances, we randomly select candidates from other sets. Therefore, the members of TS_1 can be divided into positive and negative instances (i.e. label *l* can be one or zero).

The likelihood $\mathscr{L}$ of training data can be computed as follows. In this equation, we assumed that labels *l* are generated independently.

$$\mathscr{L} = \prod_{i=1}^{|TS_1|} p_\theta\left(c = 1\middle|e, n, m, t_1, t_2\right)^{l_i} p_\theta\left(c = 0\middle|e, n, m, t_1, t_2\right)^{1-l_i},$$

in which l_i is the label of the *i*th training instance. We model $p_\theta\left(c = 1|e, n, m, t_1, t_2\right)$ by logistic functions on a linear combination of features. The unknown parameters θ can then be estimated by maximizing the following log likelihood function.

$$\begin{aligned} \theta^* \quad = \quad & \mathrm{argmax}_\theta \textstyle\sum_{i=1}^{|TS_1|} \Big(l_i \log p_\theta\left(c = 1\middle|e, n, m, t_1, t_2\right) \\ & + \left(1 - l_i\right) \log p_\theta\left(c = 0\middle|e, n, m, t_1, t_2\right)\Big) \end{aligned} \tag{10}$$

The estimated parameters can then be plugged back in $p_\theta\left(c = 1|e, n, m, t_1, t_2\right)$ to predict the topic transition probability of candidate *e*.

Although in this Section, we explain the point-wise approach using logistic regression classifier. We can also use other state of the art classification methods like SVM and decision tree to estimate the probability of $p_{t_1 \to t_2}(m|n,e)$.

In contrast with the logistic regression classifier, the output of the decision tree and SVM algorithms cannot be directly interpreted as the transition probability which we needed in Eq. (9). Therefore, for the decision tree and SVM algorithms in point-wise approach, we used the predicted binary labels to assign zero or one to $p_{t_1 \to t_2}(m|n,e)$ probability.

We refer to this proposed model in the rest of paper according to the base classifier used in the estimation of the transition probability.

5.2. Pair-wise estimation

Following the idea of pair-wise learning Liu (2009), in this Section, we propose a learning approach to estimate the probability of $p_{t_1 \to t_2}(m|n,e)$.

According to the definition of set S_4 in Section 5.1, it is possible to infer the topic preference for candidates who are experts in at least one topic at time t_1 and one topic at time t_2.

Suppose a candidate $e \in S_4$ who is expert on topic n at time t_1. We can partition the set of topics into two subset *ET* (i.e. expertise topics) and *NET* (i.e. non-expertise topics). The subset *ET* includes all topics which candidate e is expert on them at time t_2. The subset *NET* includes all topics which candidate e is not expert on them at time t_2. Intuitively, the candidate e prefers to transit to topics in set *ET* in comparison with topics in set *NET*. Specifically, for each topic pair $(m_1, m_2) \in (ET, NET)$, the following topic transition preference can be inferred:

$$m_1 <_{rank} m_2.$$

Following the idea of pair-wise learning to rank (Joachims, 2002), we can represent each quintuple (e, m, n, t_1, t_2) using a feature vector $\vec{\upsilon}$. Learning in this pair-wise approach is finding a vector $\vec{\beta}$ such that the number of rank preference constraints are maximized on the given training set. We used the Rank SVM algorithm (Joachims, 2002) to find the optimal value of $\vec{\beta}$.

After finding the optimal vector $\vec{\beta}$, we can use $\vec{\beta}\cdot \upsilon$ as a measure to determine the topic transition probability $p_{t_1 \to t_2}(m|n,e)$. We used min-max normalization to transform the $\vec{\beta}\cdot \vec{\upsilon}$ into [0,1] interval.

According to SVM Rank algorithm (Joachims, 2002), the value of $\vec{\beta}\cdot \upsilon$ can be used only for the ranking of the items according the given preferences and cannot be interpreted as the relevance probability of each item. In our problem, according to Eq. (9), we are interested in the topics (i.e. m) which the given candidate e is more likely to be expert in future time *FT*. While it is possible to use a logistic function on top of SVM rank to transform its outputs to probabilities (Platt, 1999), every monotonically increasing function of $\vec{\beta}\cdot \upsilon$ preserve the order of most probable expertise topics of a given candidate e at future time *FT*. We select the min-max normalization to transform the values of $\vec{\beta}\cdot \upsilon$ to interval of [0, 1] for three reasons. First, the min-max normalization is a parameter-free monotonically increasing function of $\vec{\beta}\cdot \upsilon$ which preserve the order of high probable expertise topics of a given candidate. Second, in comparison with the logistic normalization, the fewer number of learning parameters decreases the complexity of model and finally, utilizing the min-max normalization, it is possible to reuse the SVM rank algorithm as a black-box module in our framework without changing its learning procedure and parameters.

5.3. Feature groups

According to our observations (i.e. Figs. 1 and 2), we found four important feature groups which can affect the transition probability $p_{t_1 \to t_2}(m|n,e)$. Table 1 indicates these feature groups. We used 24 features in total which are categorized in these groups. The four feature groups includes:

1. *Topic similarity*: This feature group measures the similarity of topics m and n in estimation of $p_{t_1 \to t_2}(m|n,e)$. As mentioned before, it is more likely that a candidate changes his/her main interest from topic n to a similar topic m rather than a completely different topic.
2. *Emerging topics*: This feature group will enhance the probability of $p_{t_1 \to t_2}(m|n,e)$, if m is an emerging topic. These features indicate the popularity of emerging topic in comparison to the current topic or all topics on the basis of different features. As the popularity of each topic increases, the tendency of working on that topic increases too.
3. *User behavior*: This feature group considers the characteristics of candidate e to explore and jump to new topics. Because of the high dependency of transition probability to the user behaviors, this feature group should have the most influence in the estimation of this probability.
4. *Topic transition*: This feature group utilizes the recent transition patterns by other users and show how often this topic transition happen in the users.

The main similarity of our work with the Temporal Profile Based Model (TPBM) is that both methods relay on the Markov assumption to model the transition of a given candidate from a topic in current time to a new topic in future time. However, the

Table 1
The four feature groups used for learning of the proposed models.

Feature group	Feature name	Notation description
Topic Similarity	$f_1 = \frac{\left\|D_{m,t_1} \cap D_{n,t_1}\right\|}{\left\|D_{m,t_1} \cup D_{n,t_1}\right\|}$	D_{m,t_1} = set of post related to topic m at time t_1
	$f_2 = \frac{\left\|E_{m,t_1} \cap E_{n,t_1}\right\|}{\left\|E_{m,t_1} \cup E_{n,t_1}\right\|}$	E_{m,t_1} = set of users who publish a post related to topic m at time t_1
Emerging Topics	$f_3 = \left\|D_{m,t_1}\right\| - \left\|D_{n,t_1}\right\|$	
	$f_4 = \frac{\bar{A}_{m,t_1}}{\bar{A}_{t_1}}$	$\bar{A}_{m,t_1}$ = average answer counts of questions related to topic m at time t_1
	$f_5 = \bar{A}_{m,t_1} - \bar{A}_{n,t_1}$	
	$f_6 = \frac{\bar{C}_{m,t_1}}{\bar{C}_{t_1}}$	$\bar{C}_{m,t_1}$ = average comment counts of questions related to topic m at time t_1
	$f_7 = \bar{C}_{m,t_1} - \bar{C}_{n,t_1}$	
	$f_8 = \frac{\bar{F}_{m,t_1}}{\bar{F}_{t_1}}$	$\bar{F}_{m,t_1}$ = average favorite counts of questions related to topic m at time t_1
	$f_9 = \bar{F}_{m,t_1} - \bar{F}_{n,t_1}$	
	$f_{10} = \frac{\left\|\bar{D}_{m,t_1}\right\|}{\left\|\bar{D}_{t_1}\right\|}$	
	$f_{11} = \bar{V}_{m,t_1}$	$\bar{V}_{m,t_1}$ = average view counts of questions related to topic m at time t_1
User Behavior	$f_{12} = \left\|T_{e,t_1}\right\|$	T_{e,t_1} = set of topic which e used at time t_1
	$f_{13} = \left\|AA_{e,m,t_1}\right\|$	AA_{e,m,t_1} = set of accepted answer of user e related to topic m at time t_1
	$f_{14} = \left\|Tag_{e,m,t_1}\right\|$	Tag_{e,m,t_1} = set of tag which are related to topic m and user e used at time t_1
	$f_{15} = \left\|Tag_{e,t_1}\right\|$	Tag_{e,t_1} = set of tag which user e used at time t_1
	$f_{16} = \text{Conservativeness}$	(Fang & Godavarthy, 2014)
	$f_{17} = \left\|B_{e,t_1}\right\|$	B_{e,t_1} = set of badges which user e received at time t_1
	$f_{18} = \frac{\left\|\bar{D}_{e,m,t_1}\right\|}{\left\|\bar{D}_{e,t_1}\right\|}$	D_{e,m,t_1} = set of post related to topic m at time t_1 which are belong to user e
Topic Transition	$f_{19} = \frac{\left\|E'_{t_1-a,n,1} \cap E'_{t_1,m,1}\right\|}{\left\|E'_{t_1-a,n,1}\right\|}$	$E'_{t_1,m,1}$ = set of users who have at least one accepted answer related to m at time t_1
	$f_{20} = \frac{\left\|E'_{t_1-a,n,3} \cap E'_{t_1,m,3}\right\|}{\left\|E'_{t_1-a,n,3}\right\|}$	
	$f_{21} = \frac{\left\|E'_{t_1-a,n,10} \cap E'_{t_1,m,10}\right\|}{\left\|E'_{t_1-a,n,10}\right\|}$	
	$f_{22} = \frac{\left\|E_{t_1-a,n} \cap E_{t_1,m}\right\|}{\left\|E_{t_1-a,n}\right\|}$	$E_{t_1,m}$ = set of users who have post related to m at time t_1; a is the moving time window
	$f_{23} = \sum_{y \le t_1} \frac{\left\|E_{y-a,n} \cap E_{y,m}\right\|}{\left\|E_{y-a,n}\right\|}$	
	$f_{24} = \frac{\left\|E_{t_1,m}\right\|}{\left\|E_{t_1}\right\|}$	E_{t_1} = set of users who have post at time t_1

main differences are:

- We estimate the transition probability of a candidate form a topic at current time to a new topic in a future time using discriminative approach in contrast with the research proposed in Fang and Godavarthy (2014) which is a probabilistic generative model. Utilizing the discriminative approach is essential here because of the complexity and the variety of data in CQA in comparison with the bibliographic network data (i.e. DBLP) which is originally used in the previous research Fang and Godavarthy (2014).
- The proposed models in this Section are based on the pointwise and the pairwise learning to rank approaches Liu (2009) which are efficient and effective for large data collections.
- While the method proposed in Fang and Godavarthy (2014) focused on the prediction of a language model for a given candidate in future time, the models proposed in this Section can be used to rank experts in a CQA.

6. Experiment results

In this Section, an extensive set of experiments are designed to address the following questions of the proposed research:

- RQ1: How do the baseline models perform on future expertise problem?

- RQ2: How do point-wise and pair-wise learning models perform compared to each other?
- RQ3: What feature groups are likely more beneficial regarding future expertise ranking? Can integration of all feature groups into our proposed model improve the performance? What is the importance of each feature in our proposed model?
- RQ4: How efficient is our proposed model in each topic?

In the rest of this Section, we first introduce the experimental setup and evaluation metrics and then propose our experimental results to answer the above questions.

6.1. Experimental setup and metrics

In this Section, we describe the data collection, baseline algorithms, and evaluation metrics.

6.1.1. Data collection

Our dataset comes from Stack Overflow, which covers the period August 2008 to March 2015 and contains the information of almost 4 million users and 24 million posts (i.e., question and answer) DataDump (2015). We select a subset of the original dataset which includes all questions which have *java* as one of their tags and their associated answers. It is worth mentioning that the *java* tag is the most popular tag in our dataset. We removed other questions and their answers from the original dataset. This dataset consists of **810,071** questions, **1,510,812** answers, **18,957** tags, **270,972** distinct questioner and **206,397** distinct answerer. A more detailed statistics of our dataset is presented in Table 2.

Prior to the process of the data, we used the Lucene Standard Analyzer to remove the stop-words and to stem the words occurred in the questions and answers. In addition, the original data dump of the stack overflow includes html tags and scripts. We parsed the data and removed the html tags prior to the process of the data.

We ran MALLET topic modeling package Mallet (2016) to extract a set of 50 latent topics from the body of our post collection. In order to implement the retrieval models, we used Apache Lucene (Lucene, 2016). Similar to Fang and Godavarthy (2014), in our experiments, the time difference between current time (i.e. CT) and future time (i.e. FT) is considered to be equal to one year.

We used the top 50 most frequent tags in our dataset as the future expert finding queries. Relevant experts to each query for a given year are determined according to the number of accepted answer provided by candidates. Similar to van Dijk et al. (2015), we used an automatic approach to label experts and non-experts in our dataset. Specifically, for each pair of tags (i.e. 50 queries) and each year (9 years: 2008–2015), we defined the set of users who answered more than 10 accepted answers in that year as the experts. Table 3 indicates the list of queries we used in our experiments. According to our definition of golden measure proposed in Section 5, we randomly generate 56,036 positive and 56,194 negative training instances.

Table 2
Number of Questions(#Q), Answers(#A), Questioners(#Q user), Answerers(#A user) and the ratio of questions with accepted answer distributed over years in our test collection.

Year	#Q	#A	#Q user	#A user	Ratio of questions with accepted answer
2008	4367	16,898	1854	3955	75%
2009	24,208	77,049	7747	11,983	72%
2010	55,217	135,609	20,799	22,218	71%
2011	100,763	218,622	39,413	35,645	67%
2012	147,633	289,179	58,255	51,383	63%
2013	194,597	357,105	83,745	67,859	54%
2014	239,095	356,264	109,157	77,917	46%
2015	44,191	60,086	28,010	20,329	38%

Table 3
Top 50 most frequent tags in out test collection which are used as queries in our experiments.

Query list				
android	servlets	web service	junit	java performance
swing	string manipulation	java arraylist	database programming	java image
spring	mysql	sql	google app engine	java applet
eclipse	spring mvc	javascript	exception handeling	jframe
hibernate	java enterprise edition	socket programming	html	jtable
multithreading	json	java generic	rest	java nullpointer
java array	java persistence API	netbeans	algorithm	java methods
xml	tomcat	user interface	jsf	linux
jsp	regex	jar	gwt	java collections
maven	jdbc	file manipulation	java class concepts	jpanel

6.1.2. Evaluation metrics

Precision at rank n (*p@n*) and Mean Average Precision(*MAP*) which are two widely used IR ranking metrics, are employed to measure the performance of our proposed models. The performance measure *p@n* for a given query is the fraction of top *n* retrieved users that are experts for that query:

$$p@n = \frac{\#\text{expert users in top n retrieved results}}{n}$$

MAP is the mean value of Average Precision(*AP*) for all queries. For each query, *AP* is defined as:

$$AP = \frac{\sum_{n=1}^{N} p@n * rel(n)}{R}$$

where *N* is the total number of retrieved users, *R* is the total number of expert users and *rel*(*n*) is a binary function indicating the expertise of given candidate.

6.2. Experimental results

In this Section, we will answer the research questions mentioned in Section 6. We first compare the performance of proposed models with baseline models in Section 6.2.1 (i.e. RQ1 and RQ2). Then, we discuss the discrimination power of each feature in Section 6.2.2 (i.e. RQ3). Finally, we compare the performance of the proposed model for each topic in Section 6.2.3 (i.e. RQ4).

6.2.1. Comparison with baseline models

In this Section, we compare the performance of the proposed models with the DBM Balog et al. (2009), TBM Momtazi and Naumann (2013) and TPBM Fang and Godavarthy (2014) baseline models. In addition to these baselines, we compared our proposed models with two other baselines which are described as follows.

The proposed pairwise and pointwise models are basically discriminative approaches which utilize training data to learn the parameters and then rank the expert candidates in future time. While all above mentioned baselines are generative, we used a discriminative baseline to make the comparison fairer. In this baseline method, we do not utilize the Markov assumption indicated in Eq. (9). Specifically, we train a logistic regression classifier to directly predict the expertise score of a given candidate on a topic in future time given the value of his/her associated features in current time as indicated in the following equation. The result of $p\left(m \middle| e, CT = t_1, FT = t_2\right)$ in following equation can be plugged into Eq. (7) to rank experts in future time t_2.

$$p\left(m \middle| e, CT = t_1, FT = t_2\right) = p\left(l_{m,e,t_2} = 1 \middle| e_{t_1}\right) = \frac{1}{1 + \exp\left(\theta e_{t_1}\right)}, \tag{11}$$

In which e_{t_1} is the associated feature vector of candidate *e* at time t_1, θ is the parameter vector of the model and l_{m,e,t_2} indicates the expertise label of candidate *e* on topic *m* at time t_2. Learning in this model is finding the best value of the parameter θ which maximize the likelihood of the training data. For this baseline, we utilize the “Emerging Topics” and “User Behavior” feature groups indicated in Table 1. The “Topic Similarity” and “Topic Transition” feature groups cannot be used with this baseline because as mentioned before, we ignored the Markov assumption in this baseline model. We call this baseline as Direct LR (i.e. logistic regression) in the rest of the paper.

The next baseline is based on the TPBM which is a probabilistic model which utilize the Markov assumption. As mentioned before, the TPBM Fang and Godavarthy (2014) model is basically an expert profiling method rather than expert finding.[7] Specifically, while expert finding tries to rank all candidates for a given query (i.e. topic), expert profiling tries to recognize important expertise topics for a given candidate. The formulation of the two tasks are somehow similar and can be confusing.

As mentioned in Section 4.3, the probability of $p_{t_1}(n|e)$ indicates the strength of the association between the topic *n* and the expert *e* at time t_1. The probability of $p_{t_1}(n|e)$ is originally estimated in TPBM Fang and Godavarthy (2014) method by the ratio of the number of times the topic *n* is associated with *e* divided by number of all topics associated with *e* which is suitable for the expert profiling task. In our last baseline, we estimate the $p_{t_1}(n|e)$ probability by the ratio of the number of times the topic *n* is associated with *e* divided by the total number of times a candidate is associated with topic *n*. We call this baseline as Tuned-TPBM in the rest of the paper.

Table 4 indicates the comparison results of DBM (Balog et al., 2009), TBM (Momtazi & Naumann, 2013) and TPBM (Fang & Godavarthy, 2014),Tuned-TPBM, Direct LR, point-wise and pair-wise models.

[7] Please refer to Balog and De Rijke (2007) for comparison of the two tasks.

Table 4
Comparison of our proposed models with DBM (Balog et al., 2009), TBM (Momtazi & Naumann, 2013) and TPBM (Fang & Godavarthy, 2014) improvements are indicated against the best baseline. * indicates the improvement is statistically significant ($\rho < 0.05$).

Method	P@1	P@5	P@10	MAP
DBM(All) (Balog et al., 2009)	0.251	0.280	0.216	0.197
DBM(Recent) (Balog et al., 2009)	0.423	0.346	0.261	0.280
TBM (Momtazi & Naumann, 2013)	0.543	0.515	0.409	0.499
TPBM (Fang & Godavarthy, 2014)	0.072	0.061	0.039	0.057
TPBM-Tuned (Section 6.2.1)	0.480	0.430	0.331	0.390
Direct LR (Section 6.2.1)	0.852	0.555	0.239	0.511
Point-wise Model(LR)	0.866	0.582	0.394	0.536
Improvement vs Direct LR (%)	1.6%	4.9%*	64.9%*	4.9%*
Point-wise Model (Decision Tree (C5.0))	0.900	0.612	0.421	0.587
Improvement vs Direct LR (%)	5.6%*	10.3%*	76.2%*	14.9%*
Point-wise Model(SVM)	**0.952**	**0.679**	**0.483**	**0.697**
Improvement vs Direct LR (%)	11.7%*	22.3%*	102.1%*	36.4%*
Pairwise Model	0.790	0.572	0.414	0.529
Improvement vs Direct LR (%)	-7.3%	3.1%	73.2%*	3.5%*

The performance results of the document based models are reported in two different cases: *all* and *recent* DBM models. As explained in Section 4.1, while *all* approach uses all associated documents of a candidate to estimate his/her expertise, the *recent* approach uses the recent documents (in our experiments we set the parameter a to be equal to one year). According to Table 4, considering only recent documents of a candidate to predict his/her expertise can improve all performance measures in comparison with the *all* approach. This result confirms that the future related topics to a candidate are highly dependent to his/her recent related topics of expertise. Intuitively, this observation indicates that the Markov assumption which we used as the main principal in our retrieval model is valid and is confirmed by the empirical experiment.

The result reported in Table 4, indicates the topic based model (i.e. TBM) (Momtazi & Naumann, 2013) has better performance in comparison with the DBM models. Utilizing the latent topics, the TBM model can reduce the vocabulary gap between query and the documents and accordingly improve the overall performance of retrieval. As explained in Section 4.2, similar to TBM (Momtazi & Naumann, 2013), our proposed models are also basically topic based retrieval models. This result of TBM model confirms that the topic based retrieval model is a rational choice for the future expertise problem.

According to Table 4, the performance of the temporal profile based model (i.e. TPBM) (Fang & Godavarthy, 2014) is surprisingly very low in our test collection. We found two main reasons for this observation. First, the temporal profile based model (i.e. TPBM) (Fang & Godavarthy, 2014) is an *expert profiling* method which predicts the language model of each candidate in future given evidence in current time. As suggested by Balog and De Rijke (2007), the models specifically designed for expert profiling can not be used directly for expert finding and vise versa. To be more specific, expert profiling approaches like TPBM (Fang & Godavarthy, 2014) addresses the estimation of profile of each candidate in-isolation and cannot be used directly for expert finding which emphasizes on ranking of candidates in comparison to each other. Second, the TPBM model only considers the user conservativeness, similarity, and popularity of topics to model the dynamics of expertise which seems not enough to capture the dynamicity and complexity of data at StackOverflow. In contrast, the TPBM-Tuned model described in Section 6.2.1 is basically an expert finding model which has better performance than the TPBM but because of the complexity of data in our dataset it can not perform better than the TBM model. Finally, our last baseline model is the discriminative model described in Section 6.2.1 which has better performance in comparison with other generative models.

Table 4 indicates the results of our proposed models and compares them with the baselines. As we can see in Table 4, our proposed models outperform baselines regarding all performance measures. As mentioned before, we implement point-wise approach by using three classification methods namely logistic regression, decision tree (C5.0) and SVM (with Gaussian kernel). As shown in Table 4, the point-wise model which uses SVM classifier has the best overall performance. Although the pair-wise model outperforms baselines, it shows slightly fewer performance compared to the point-wise models. It is worth mentioning that our proposed discriminative models outperforms the Direct LR (Section 6.2.1) model. It means that the Markov property and the related features extracted based on this assumption can help to predict future experts on a given topic.

6.2.2. The discriminative power of feature groups

In this Section, we analyze the discriminative power of each feature group in the task of Future Expert Finding. Because of the higher performance of point-wise models in comparison to pair-wise model regarding future expertise ranking, we only consider these models and trained them on each feature group separately.

Table 5 reports the performance results of three point-wise models which use logistic regression, decision tree, and SVM classifiers. These models are trained on each feature group separately. Also, the result of the corresponding model which is trained using all feature groups is also reported.

Table 5
Comparison of point-wise models for each feature group.

Model name	Feature group	P@1	P@5	P@10	MAP
Logistic Regression	Topic Similarity	0.182	0.113	0.077	0.110
	User Behavior	0.856	0.566	0.383	0.519
	Topic Transition	0.206	0.118	0.080	0.131
	Emerging Topics	0.113	0.069	0.045	0.083
	All	**0.866**	**0.582**	**0.394**	**0.536**
SVM (Gaussian Kernel)	Topic Similarity	0.289	0.142	0.086	0.100
	User Behavior	**0.959**	**0.685**	0.483	**0.710**
	Topic Transition	0.378	0.249	0.179	0.198
	Emerging Topics	0.076	0.044	0.027	0.038
	All	0.952	0.679	**0.483**	0.697
Decision Tree (C5.0)	Topic Similarity	0.320	0.170	0.103	0.121
	User Behavior	**0.907**	**0.616**	**0.429**	**0.603**
	Topic Transition	0.574	0.392	0.275	0.326
	Emerging Topics	0.227	0.130	0.085	0.077
	All	0.900	0.612	0.421	0.587

According to this experiment, the *User Behavior* and *Topic Transition* feature groups are more effective than the two other feature groups. Interestingly, training the model only by using the *User Behavior* feature group has almost the same performance in comparison with models trained by all feature groups.

We also examine the importance of each feature in one of our proposed model which has the best overall performance (i.e. point-wise model which uses SVM classifier). We compute the area under the ROC curve for each feature. Then we can consider this area as the importance score of each feature. Fig. 3 indicates the importance score of each feature in point-wise (SVM)

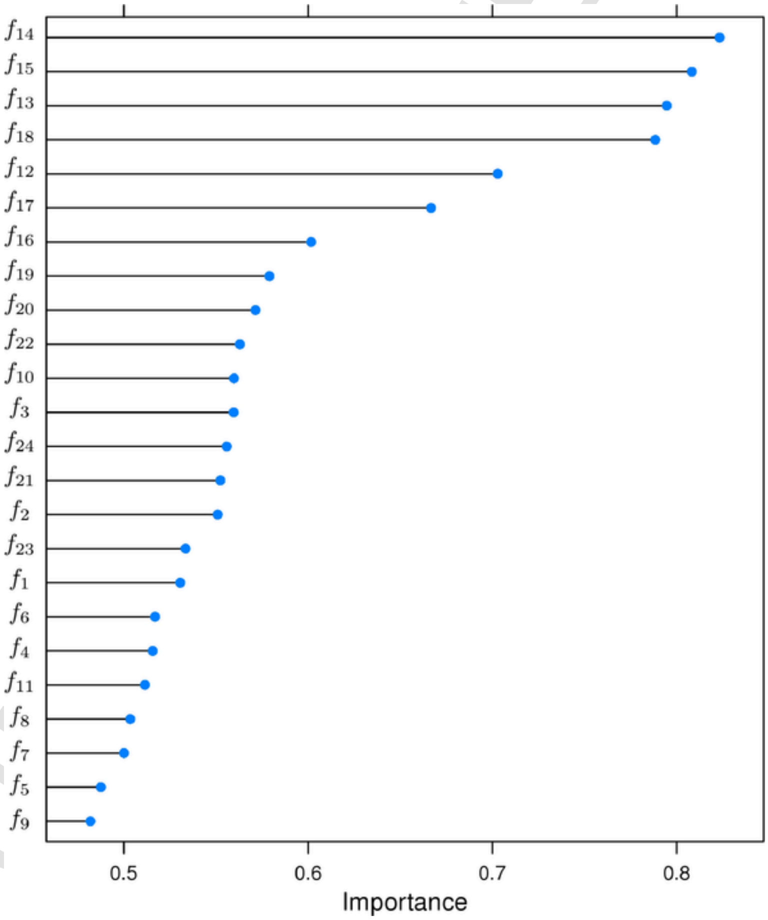

Fig. 3. Comparison of importance of each feature in point-wise model(SVM).

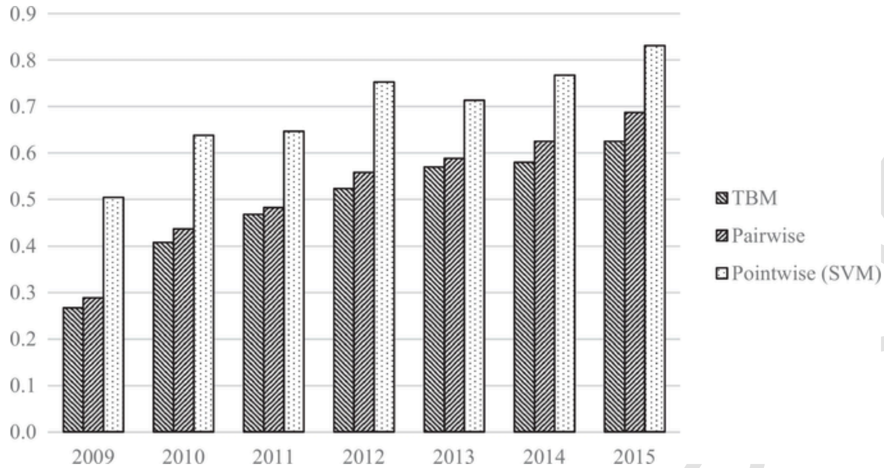

Fig. 4. Comparison of MAP of our proposed models with TBM model Momtazi and Naumann (2013) for each year.

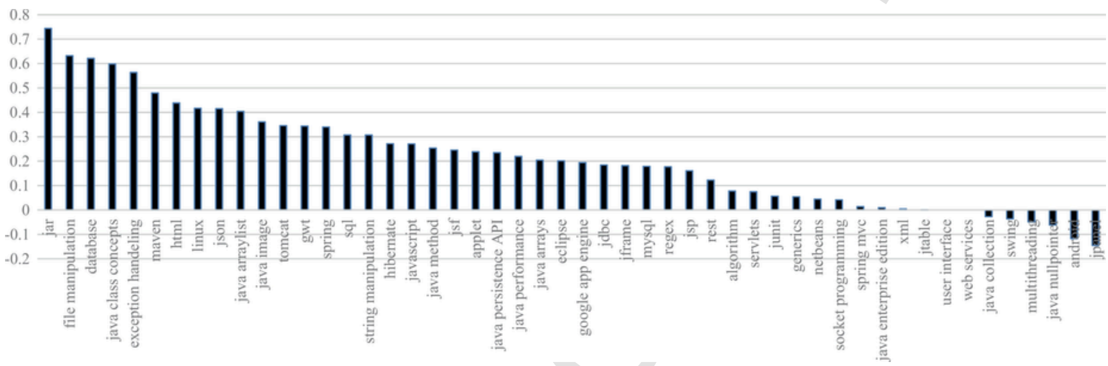

Fig. 5. Comparison of MAP of our proposed model pointwise (SVM) with TBM model (Momtazi & Naumann, 2013) for each query.

model. According to this Figure, user behaviors have the most influence between all of the proposed features. This observation is in agreement with previous results because of the high dependency of the topic transition probability with the user activities. By omitting the user from the predicting features, the performance of the models drops down. The poor performance of Emerging Topic group in the Table 5 and Fig. 3 can prove this reasoning.

6.2.3. Detailed comparison

Fig. 4 indicates the performance of point-wise (SVM), pair-wise and TBM model Momtazi and Naumann (2013) for each year. According to this Figure, the overall performance of the proposed retrieval models are increasing each year. This observation can be explained by increasing amount of data available for training for each year. Specifically, the lowest performance measures in 2009 can be explained by the training data sparsity in the year 2008 because the Stack Overflow started at 2008 and the number of users and interactions are lower in this year in comparison to other years. The main observation in this experiment is that independent of the query time the performance of the point-wise method is significantly better than our best baseline model.

Fig. 5 indicates the MAP difference between point-wise (SVM) and TBM model (Momtazi & Naumann, 2013) for each query. Except for some few queries, the mean average precision of the proposed model is significantly better than the TBM model (Momtazi & Naumann, 2013).

7. Conclusion and future work

In this paper, we investigate the problem of *Future Expert Finding* which focuses on ranking of experts in future while the expertise evidence is observed in the current time. We proposed a supervised learning framework for this problem and also automatically generated a training and evaluation dataset for this problem. We evaluated the performance of our proposed model on StackOverflow which is a well known community question answering website. We also proposed four feature groups which can beneficial to the prediction of the expertise topics of a candidate. Out proposed models are based on the Markov assumption which means that the future expertise topics of a candidate depends on the his/her current topics of expertise. The result of our experiments indicates that the Markov assumption can be beneficial to predict the expertise topics of a candidate. We designed point-wise and pair-wise learning to rank models based on this assumption. Our experiment indicated that the point-wise approach can outperform all the baseline models as well as the pair-wise modeling. In this research, we used the number of accepted answer of users as a measure to label these expert users. In the next step, we plan to generalize our solution to predict not only the experts but also the valuable users (i.e. high reputed users) in CQA, which can have great value to improve the quality of user experience and engagement.

References

Alarfaj, F., Kruschwitz, U., Hunter, D., Fox, C., 2012. Finding the right supervisor: Expert-finding in a university domain. Proceedings of the 2012 conference of the North American chapter of the association for computational linguistics: Human language technologies: Student research workshop. Association for Computational Linguistics, 1–6.

Apache lucene (2016). https://lucene.apache.org/core/. [Online; accessed 18-August-2016].

Asaduzzaman, M., Mashiyat, A.S., Roy, C.K., Schneider, K.A., 2013. Answering questions about unanswered questions of stack overflow. MSR '13, 97–100.

Balog, K., Azzopardi, L., de Rijke, M., 2009. A language modeling framework for expert finding. Information Processing & Management 45 (1), 1–19.

Balog, K., Bogers, T., Azzopardi, L., de Rijke, M., van den Bosch, A., 2007. Broad expertise retrieval in sparse data environments. SIGIR '07, 551–558.

Balog, K., De Rijke, M., 2007. Determining expert profiles (with an application to expert finding). IJCAI'07, San Francisco, CA, USA, 2657–2662.

Balog, K., Fang, Y., de Rijke, M., Serdyukov, P., Si, L., 2012. Expertise retrieval. Foundations and Trends in Information Retrieval 6, 127–256.

Blei, D.M., Ng, A.Y., Jordan, M.I., 2003. Latent Dirichlet allocation. Journal of Machine Learning Research 3, 993–1022.

Budalakoti, S., Bekkerman, R., 2012. Bimodal invitation-navigation fair bets model for authority identification in a social network. Proceedings of the 21st international conference on world wide web. ACM, 709–718.

Daud, A., Ahmad, M., Malik, M.S.I., Che, D., 2015. Using machine learning techniques for rising star prediction in co-author network. Scientometrics 102 (2), 1687–1711.

Daud, A., Li, J., Zhou, L., Muhammad, F., 2009. Exploiting temporal authors interests via temporal-author-topic modeling. Proceedings of the international conference on advanced data mining and applications. Springer, 435–443.

Daud, A., Li, J., Zhou, L., Muhammad, F., 2010. Temporal expert finding through generalized time topic modeling. Knowledge-Based Systems 23 (6), 615–625.

Deng, H., Han, J., Lyu, M.R., King, I., 2012. Modeling and exploiting heterogeneous bibliographic networks for expertise ranking. Proceedings of the 2012 ACM/IEEE joint conference on digital libraries (JCDL 2012). IEEE.

Deng, H., King, I., Lyu, M.R., 2008. Formal models for expert finding on DBLP bibliography data. Proceedings of the eighth IEEE international conference on data mining. IEEE Computer Society, 163–172.

Fang, Y., Godavarthy, A., 2014. Modeling the dynamics of personal expertise. SIGIR '14, 1107–1110.

Furtado, A., Andrade, N., Oliveira, N., Brasileiro, F., 2013. Contributor profiles, their dynamics, and their importance in five Q&A sites. Proceedings of the 2013 conference on computer supported cooperative work. ACM, 1237–1252.

Hashemi, S.H., Neshati, M., Beigy, H., 2013. Expertise retrieval in bibliographic network: A topic dominance learning approach. CIKM '13, 1117–1126.

Joachims, T., 2002. Optimizing search engines using clickthrough data. KDD '02, 133–142.

Kargar, M., An, A., 2011. Discovering top-k teams of experts with/without a leader in social networks. CIKM '11, New York, NY, USA, 985–994.

Karimzadehgan, M., White, R.W., Richardson, M., 2009. Enhancing expert finding using organizational hierarchies. Proceedings of the 31th European conference on ir research on advances in information retrieval. Springer-Verlag, 177–188.

Lappas, T., Liu, K., Terzi, E., 2009. Finding a team of experts in social networks. Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, 467–476.

Li, H., Jin, S., Shudong, L., 2015. A hybrid model for experts finding in community question answering. Proceedings of the 2015 international conference on cyber-enabled distributed computing and knowledge discovery (cyberc). IEEE, 176–185.

Liang, S., de Rijke, M., 2016. Formal language models for finding groups of experts. Information Processing & Management 52 (4), 529–549.

Linares-Vásquez, M., Bavota, G., Di Penta, M., Oliveto, R., Poshyvanyk, D., 2014. How do api changes trigger stack overflow discussions? A study on the android sdk. ICPC 2014, 83–94.

Liu, T.-Y., 2009. Learning to rank for information retrieval. Foundations and Trends in Information Retrieval 3 (3), 225–331.

Mallet (2016). http://mallet.cs.umass.edu. [Online; accessed 18-August-2016].

McAuley, J.J., Leskovec, J., 2013. From amateurs to connoisseurs: Modeling the evolution of user expertise through online reviews. Proceedings of the 22nd international conference on world wide web. ACM, 897–908.

Momtazi, S., Naumann, F., 2013. Topic modeling for expert finding using latent Dirichlet allocation. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 3 (5), 346–353.

Mukherjee, S., Lamba, H., Weikum, G., 2015. Experience-aware item recommendation in evolving review communities. Proceedings of the 2015 IEEE international conference on data mining (ICDM). IEEE, 925–930.

Neshati, M., Asgari, E., Hiemstra, D., Beigy, H., 2013. A joint classification method to integrate scientific and social networks. European conference on information retrieval. Springer, 122–133.

Neshati, M., Beigy, H., Hiemstra, D., 2012. Multi-aspect group formation using facility location analysis. Proceedings of the seventeenth australasian document computing symposium. ACM, 62–71.

Neshati, M., Beigy, H., Hiemstra, D., 2014. Expert group formation using facility location analysis. Information Processing & Management 50 (2), 361–383.

Neshati, M., Hashemi, S.H., Beigy, H., 2014. Expertise finding in bibliographic network: Topic dominance learning approach. IEEE Transactions on Cybernetics 44 (12), 2646–2657.

Neshati, M., Hiemstra, D., Asgari, E., Beigy, H., 2014. Integration of scientific and social networks. World Wide Web 17 (5), 1051–1079.

Pal, A., 2015. Metrics and algorithms for routing questions to user communities. ACM Transactions on Information and System 33 (3), 14:1–14:29.

Pal, A., Chang, S., Konstan, J.A., 2012. Evolution of experts in question answering communities.. Proceedings of the international AAAI conference on web and social media (ICWSM).

Pal, A., Farzan, R., Konstan, J.A., Kraut, R.E., 2011. Early detection of potential experts in question answering communities. UMAP'11, 231–242.

Pal, A., Harper, F.M., Konstan, J.A., 2012. Exploring question selection bias to identify experts and potential experts in community question answering. ACM Transactions on Information and System 30 (2), 10:1–10:28.

Pal, A., Harper, F.M., Konstan, J.A., 2012. Exploring question selection bias to identify experts and potential experts in community question answering. ACM Transactions on Information Systems (TOIS) 30 (2), 10.

Pal, A., Herdagdelen, A., Chatterji, S., Taank, S., Chakrabarti, D., 2016. Discovery of topical authorities in instagram. Proceedings of the 25th international conference on world wide web. 1203–1213.

Patil, S., Lee, K., 2016. Detecting experts on quora: By their activity, quality of answers, linguistic characteristics and temporal behaviors. Social Network Analysis and Mining 6 (1), 1–11.

Platt, J.C., 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Advances in large margin classifiers. MIT Press, 61–74.

Ravi, S., Pang, B., Rastogi, V., Kumar, R., 2014. Great question! Question quality in community Q&A. Proceedings of the eighth international conference on weblogs and social media, ICWSM 2014, Ann Arbor, Michigan, USA, june 1–4, 2014..

Riahi, F., Zolaktaf, Z., Shafiei, M., Milios, E., 2012. Finding expert users in community question answering. Proceedings of the 21st international conference on world wide web. 791–798.

Rowe, M., 2013. Changing with time: Modelling and detecting user lifecycle periods in online community platforms. Proceedings of the international conference on social informatics, Springer, 30–39.

Rowe, M., 2013. Mining user lifecycles from online community platforms and their application to churn prediction. Proceedings of the 2013 IEEE 13th international conference on data mining, 637–646.

Rowe, M., 2016. Mining user development signals for online community churner detection. ACM Transactions on Knowledge Discovery from Data (TKDD) 10 (3), 21.

Rybak, J., Balog, K., Nørvåg, K., 2014. Temporal expertise profiling. Proceedings of the European conference on information retrieval. Springer, 540–546.

Srba, I., Grznar, M., Bielikova, M., 2015. Utilizing non-QA data to improve questions routing for users with low QA activity in CQA. Proceedings of the 2015 IEEE/ACM international conference on advances in social networks analysis and mining 2015. ACM, 129–136.

Stach exchange data dump (2015). https://archive.org/details/stackexchange.[Online; accessed 1-September-2015].

Stack overflow candidate search (2016). http://business.stackoverflow.com/careers/us/platform/candidate-search. [Online; accessed 18-August-2016].

Stack overflow job listings (2016). http://business.stackoverflow.com/careers/us/platform/job-listings. [Online; accessed 18-August-2016].

Sung, J., Lee, J.-G., Lee, U., 2013. Booming up the long tails: Discovering potentially contributive users in community-based question answering services.. Proceedings of the international AAAI conference on web and social media (ICWSM).

Toba, H., Ming, Z.-Y., Adriani, M., Chua, T.-S., 2014. Discovering high quality answers in community question answering archives using a hierarchy of classifiers. Information Sciences 261, 101–115.

van Dijk, D., Tsagkias, M., de Rijke, M., 2015. Early detection of topical expertise in community question answering. SIGIR '15, 995–998.

Xu, Z., Ramanathan, J., 2016. Thread-based probabilistic models for expert finding in enterprise microblogs. Expert Systems with Applications 43, 286–297.

Yang, J., Tao, K., Bozzon, A., Houben, G.-J., 2014. Sparrows and owls: Characterisation of expert behaviour in stackoverflow. Proceedings of the international conference on user modeling, adaptation, and personalization. Springer, 266–277.

Yao, Y., Tong, H., Xie, T., Akoglu, L., Xu, F., Lu, J., 2015. Detecting high-quality posts in community question answering sites. Information Sciences 302, 70–82.

Yeniterzi, R., Callan, J., 2015. Moving from static to dynamic modeling of expertise for question routing in CQA sites. Proceedings of the ninth international AAAI conference on web and social media.

Zhai, C., Lafferty, J., 2004. A study of smoothing methods for language models applied to information retrieval. ACM Transactions on Information and System 22 (2).

Zhang, J., Tang, J., Li, J., 2007. Expert finding in a social network. Proceedings of the international conference on database systems for advanced applications. Springer, 1066–1069.

Zhou, Y., Cong, G., Cui, B., Jensen, C.S., Yao, J., 2009. Routing questions to the right users in online communities. Proceedings of the 2009 IEEE 25th international conference on data engineering. 700–711.

Ziaimatin, H., Groza, T., Bordea, G., Buitelaar, P., Hunter, J., 2014. Expertise profiling in evolving knowledge curation platforms. GSTF Journal on Computing (JoC) 2 (3).