SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

November 2019, Vol. 62 212103:1–212103:19
https://doi.org/10.1007/s11432-018-9848-2

IEA: an answerer recommendation approach on stack overflow

Liting WANG, Li ZHANG & Jing JIANG*

State Key Laboratory of Software Development Environment, Beihang University, Beijing 100191*, China*

Received 10 September 2018/Revised 19 October 2018/Accepted 28 February 2019/Published online 18 September 2019

Abstract Stack overflow is a web-based service where users can seek information by asking questions and share knowledge by providing answers about software development. Ideally, new questions are assigned to experts and answered within a short time after their submissions. However, the number of new questions is very large on stack overflow, answerers are not easy to find suitable questions timely. Therefore, an answerer recommendation approach is required to assign appropriate questions to answerers. In this paper, we make an empirical study about developers' activities. Empirical results show that 66.24% of users have more than 30% of comment activities. Furthermore, active users in the previous day are likely to be active in the next day. In this paper, we propose an approach IEA which combines user topical interest, topical expertise and activeness to recommend answerers for new questions. We first model user topical interest and expertise based on historical questions and answers. We also build a calculation method of users' activeness based on historical questions, answers, and comments. We evaluate the performance of IEA on 3428 users containing 41950 questions, 64894 answers, and 96960 comments. In comparison with the state-of-the-art approaches of TEM, TTEA and TTEA-ACT, IEA improves nDCG by 2.48%, 3.45% and 3.79%, and improves Pearson rank correlation coefficient by 236.20%, 84.91% and 224.12%, and improves Kendall rank correlation coefficient by 424.18%, 1845.30% and 772.60%.

Keywords answerer recommendation, activeness, comments, topical interest, topical expertise, stack overflow

Citation Wang L T, Zhang L, Jiang J. IEA: an answerer recommendation approach on stack overflow. Sci China Inf Sci, 2019, 62(11): 212103, https://doi.org/10.1007/s11432-018-9848-2

1 Introduction

Community question answering (CQA) services allow users to ask and answer questions on the web. The CQA services such as stack overflow provide an online platform for users to post their questions and share their knowledge by answering questions from others. The stack overflow[1] focuses on questions and answers related to programming, and it is popular for software developers. With the rapidly increasing number of new questions on stack overflow, an answerer recommendation approach is required to assign new questions to suitable answerers.

There have been several studies about answerer recommendation. Some previous studies [1–3] analyzed past questions and answers to recommend answerers in CQA websites. These existing approaches consider user topical interest and expertise to recommend answerers. Other previous work [4] analyzed user activeness to recommend answerers in CQA websites. In CQA websites, users not only post questions and answers in CQA websites, but also post comments on these questions and answers. In existing

* Corresponding author (email: jiangjing@buaa.edu.cn)
[1] http://stackoverflow.com/.

work [4], user activeness is analyzed based on historical questions and answers, without considering comments. Besides, existing answerer recommendation approaches only consider user topical interest and expertise [3, 4], or user topical interest and activeness [4]. However, they do not combine topical interest, topical expertise and activeness together to recommend answerers.

In this paper, we first make an empirical study about developers' activities. Results show that 66.24% of users have more than 30% of comment activities. Comments are important activities of users. Furthermore, results show that active users in the previous day are likely to be active in the next day. Based on the results of our empirical study, we build a calculation method of users' activeness by historical questions, answers and comments. Moreover, we use topic expertise model (TEM) proposed in [3] to model user topical interest and expertise based on historical questions and answers, without comments. Finally, we propose an approach topical interest, topical expertise and activeness (IEA) which combines user topical interest, topical expertise and activeness to recommend answerers.

We evaluate our approach by collecting datasets from stack overflow. Our datasets include 3428 users, 41950 questions, 64894 answers and 96960 comments. The performance of the approach is measured in terms of normalized discounted cumulative gain (nDCG), Pearson rank correlation coefficient and Kendall rank correlation coefficient [3]. The experimental results show that IEA has good performance in answerer recommendation. Specifically, IEA achieves nDCG@1, nDCG@5, nDCG, Pearson rank correlation coefficient and Kendall rank correlation coefficient values of 0.6624, 0.8349, 0.9020, 0.1880 and 0.1649, respectively. Besides, in comparison with the approaches of TEM [3], temporal topic expertise activity (TTEA) [4] and TTEA-activeness level (TTEA-ACT) [4], IEA improves nDCG@1 by 10.29%, 14.53% and 15.17%, and improves nDCG@5 by 2.68%, 3.74% and 4.11%, and improves nDCG by 2.48%, 3.45% and 3.79%, and improves Pearson rank correlation coefficient by 236.20%, 84.91% and 224.12%, and improves Kendall rank correlation coefficient by 424.18%, 1845.30% and 772.60%. Besides, we analyze the effect of the number of topics and expertise to answerer recommendation. Results show that when the number of topics and the number of expertise are all set as 10, IEA has the best performance in answerer recommendation.

The main contributions of the paper are as follows:

- We find that comments are important activities of users, and active users in the previous day are likely to be active in the next day.
- We build a calculation method of users' activeness based on historical questions, answers and comments.
- We propose an approach IEA which combines user topical interest, topical expertise and activeness to recommend answerers.
- Results show that IEA outperforms TEM [3], TTEA [4] and TTEA-ACT [4] by substantial margins.

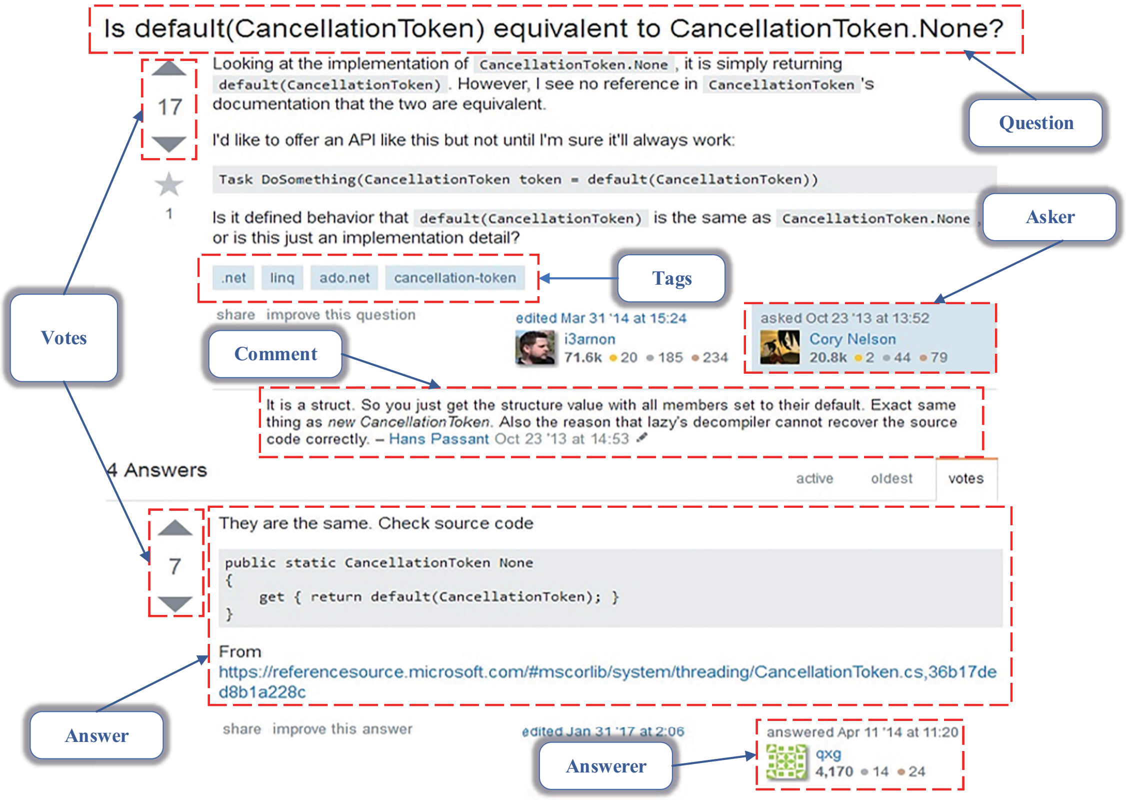
The reminder of the paper is organized as follows. Section 2 presents background and data collection. Section 3 describes our answerer recommendation approach IEA. Experiment results are shown in Section 4. Section 5 discusses threats to validity, and Section 6 introduces related studies. Finally, Section 7 concludes this paper.

2 Background and data collection

In this section, we begin by providing background information on stack overflow. Then, we introduce how datasets are collected, and we report statistics of our datasets.

2.1 Background

Stack overflow is one of the biggest question answering sites in which users share knowledge and seek expert advices on a wide range of topics in computer programming. There are more than 14 million questions on stack overflow, and most of the questions are generally related to the programming problems [2]. Users on stack overflow can ask and answer questions, and comment on questions and answers. When users ask questions on stack overflow, they need to manually assign tags to questions. Users can vote up



Is default(CancellationToken) equivalent to CancellationToken.None?

17

Looking at the implementation of `CancellationToken.None`, it is simply returning `default(CancellationToken)`. However, I see no reference in `CancellationToken`'s documentation that the two are equivalent.

I'd like to offer an API like this but not until I'm sure it'll always work:

```
Task DoSomething(CancellationToken token = default(CancellationToken))
```

1

Is it defined behavior that `default(CancellationToken)` is the same as `CancellationToken.None`, or is this just an implementation detail?

.net linq ado.net cancellation-token

share improve this question

edited Mar 31 '14 at 15:24
i3arnon
71.6k ● 20 ● 185 ● 234

asked Oct 23 '13 at 13:52
Cory Nelson
20.8k ● 2 ● 44 ● 79

It is a struct. So you just get the structure value with all members set to their default. Exact same thing as *new CancellationToken*. Also the reason that lazy's decompiler cannot recover the source code correctly. – Hans Passant Oct 23 '13 at 14:53

4 Answers

active oldest votes

7

They are the same. Check source code

```
public static CancellationToken None
{
    get { return default(CancellationToken); }
}
```

From https://referencesource.microsoft.com/#mscorlib/system/threading/CancellationToken.cs,36b17ded8b1a228c

share improve this answer

edited Jan 31 '17 at 2:06

answered Apr 11 '14 at 11:20
qxg
4,170 ● 14 ● 24

Figure 1 (Color online) An example of a question on stack overflow.

or vote down on the questions and answers. The vote score is the number of positive (up) votes minus the number of negative (down) votes. For example, a question has 5 positive votes and 3 negative votes. This vote score of this question is 2 (5−3). The vote score of answer is computed in the same way.

To illustrate the contribution process, an example of a question with number 23011187[2] which is answered by a user named qxg is shown in Figure 1. In order to make the figure clear, we show main content of this question, and delete some comments and answers. First, a user named Cory Nelson asks "Is default (CancellationToken) equivalent to CancellationToken.None". Second, the question is attached with a set of tags, including .net, linq, ado.net and cancellation-token. Moreover, a user named Hans Passant comments the question. Third, the user qxg answers the question. Finally, the vote score of the question is 17, and the vote score of answer for user qxg is 7.

2.2 Data collection

Stack overflow is the most popular question answering community focusing on computer programming. The data of stack overflow is publicly available datasets provided by creative commons data dump service[3]. It provides us valuable opportunities for research.

We download the complete datasets which are launched from January 1st 2014 to July 31th 2014 on stack overflow. We choose datasets in 2014, because these datasets are stable. Some answers still receive votes a long time after their creation, and their vote scores is changed. For questions and answers, we extract their ID, type ID (1 for question, 2 for answer), user ID, creation time, scores, tags, and their body. For each comment, we extract its creation time, user ID, question/answer ID. According to previous work [3], we select all users who have more than 80 times of questions and answers as our datasets, and obtain 3428 users. We select all questions, answers and comments posted by 3428 users from January

[2] http://stackoverflow.com/questions/19543623/is-defaultcancellationtoken-equivalent-to-cancellationtoken-none/23011187#23011187.

[3] http://archive.org/details/stackexchange.

Table 1 Number of answers per question in training data

The number of answers per question	The number of questions
0	3053
1	20423
2	10074
3	4066
4	1640
5	646
6	274
7	125
8	67
9	31
$\geqslant 10$	56

Table 2 An example of user activity in April 2014 on stack overflow

User ID	Activity	Creation time	Question ID
3523446	Answer	2014-04-11 11:20	23011187
3523446	Answer	2014-04-13 04:31	23039131
3523446	Answer	2014-04-13 04:36	23039155
3523446	Comment	2014-04-13 05:47	23039155
3523446	Answer	2014-04-13 06:16	23039802
3523446	Answer	2014-04-24 12:57	23269620
3523446	Answer	2014-04-24 13:23	23270226
3523446	Comment	2014-04-24 13:29	23269620
3523446	Answer	2014-04-25 04:38	23284281
3523446	Answer	2014-04-25 09:47	23289638
3523446	Answer	2014-04-25 12:02	23292561
3523446	Comment	2014-04-25 12:32	23284281
3523446	Comment	2014-04-25 15:16	23284281
3523446	Comment	2014-04-25 15:25	23292561
3523446	Comment	2014-04-25 16:04	23284281
3523446	Comment	2014-04-26 02:24	23305872

1st 2014 to June 30th 2014 as training data. In training data, we have 40455 questions, 61072 answers and 93054 comments posted by 3428 users. Moreover, we count the number of answers to per question in training data, as shown in Table 1. The number of questions with one answer is 20423, and the number of questions with two answers is 10074. The majority of questions have less than 3 answers. In testing data, we remove questions which have less than 2 answers according to previous studies [3]. Furthermore, testing data includes 1495 questions, 3822 answers and 3906 comments posted by 3428 users from July 1st 2014 to July 31th 2014. In total, our datasets include 41950 questions, 64894 answers and 96960 comments posted by 3428 users from January 1st 2014 to July 31th 2014 on stack overflow.

3 Proposed approach

In this section, we describe our approach IEA which combines user topical interest, topical expertise and activeness to recommend answerers. We first introduce motivation and framework of our approach, respectively. Then, we describe details of our answerer recommendation approach.

3.1 Motivation

In Subsection 2.1, Figure 1 shows an example of a question with number 23011187 which is answered by a user named qxg. We take a further step and describe more examples of the user. Table 2 describes basic

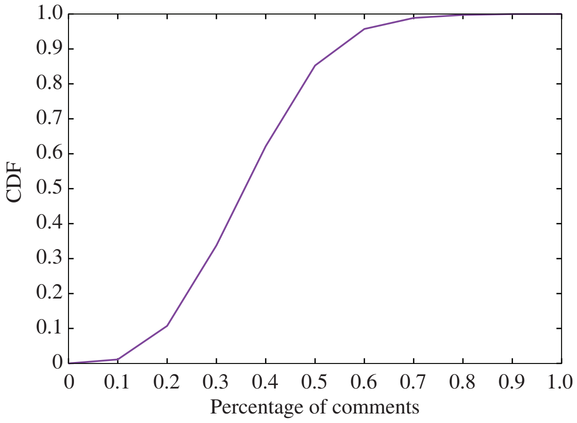


Figure 2 (Color online) Percentage of comment activities.

information of user named qxg in April 2014. In Table 2, user qxg provides 9 answers and 7 comments. Comments cover 43.75% of activities for user qxg in April 2014. Comments are important activities for user qxg. Furthermore, user qxg is active in two periods. The first period is between April 11th 2014 and April 13th 2014, and the second period is between April 24th 2014 and April 26th 2014. When user qxg is active, this user may take other actions in recent time.

We wonder whether comments are also important for other users. For each user, we compute the number of his/her comments, divided by the number of his/her questions, answers and comments. We then aggregate across all users the proportion of comment activities, and plot cumulative distribution function (CDF) in Figure 2. Figure 2 shows that 33.76% of users have less than 30% of comment activities, but the other 66.24% of users have more than 30% of comment activities. Results show that comments are important activities of users. Therefore, comments may be used to compute the activeness of users in answerer recommendation.

We further make statistic 3428 users from January 1st 2014 to June 30th 2014. First, we collect questions, answers and comments posted by 3428 users. Second, we decide whether the users are active on a certain day based on the crawled data. If a user has any questions, answers or comments, this user is considered as active, otherwise this user is considered as inactive. Third, we build the set U_i of users who are active on the day i. Fourth, the active ratio R_i on the day i is computed as the intersection of U_i and U_{i-1}, divided by U_i. The intersection of U_i and U_{i-1} includes users who are both active on day i and day $i-1$. The active ratio R_i shows the percentage of active users on day i who are also active on day $i-1$. Finally, we plot the active ratio from January 1st 2014 to June 30th 2014 in Figure 3. We find that 82.11% of active users on the day 22 are also active on the day 21. In 175 days, 84.57% of days (148 days) have active ratio higher than 60%. Active users in the previous day are likely to be active in the next day. Due to rests on weekends, active ratio decreases greatly in almost every 7 days. Although the number of active users on weekends is greatly reduced, active ratios are still more than 45% on weekends.

3.2 Overall framework

As shown in Figure 4, the overall framework contains two phases: model construction phase and recommendation phase. In model construction phase, our goal is to build a model from historical questions, answers and comments. In recommendation phase, the model is used to recommend answerers.

In model construction phase, our framework first collects various information from a set of historical questions, answers and comments. We first model user topical interest and topical expertise by using TEM proposed in [3] based on historical questions and answers. In Subsection 3.3, we describe detailed definitions of topical interest and topical expertise and why we choose them. Moreover, we build user activeness by questions, answers and comments. We describe detailed definition of activeness and why we choose activeness in Subsection 3.4.

In recommendation phase, IEA is used to recommend answerers for new questions. For each new

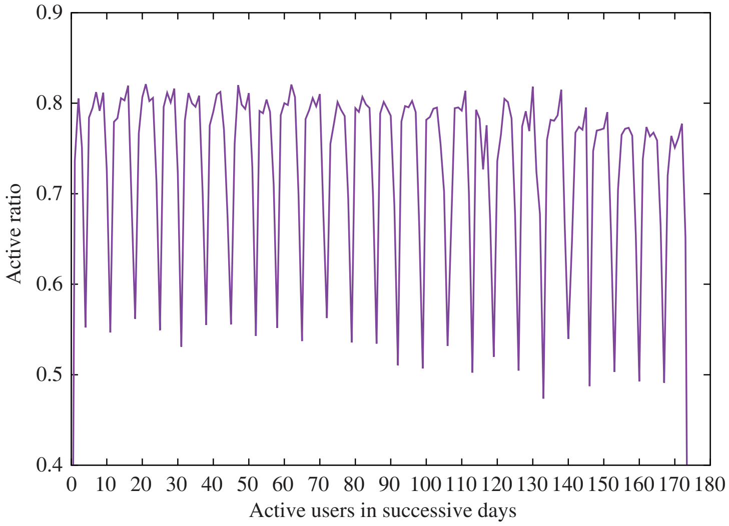


Figure 3 (Color online) Active users in successive days.

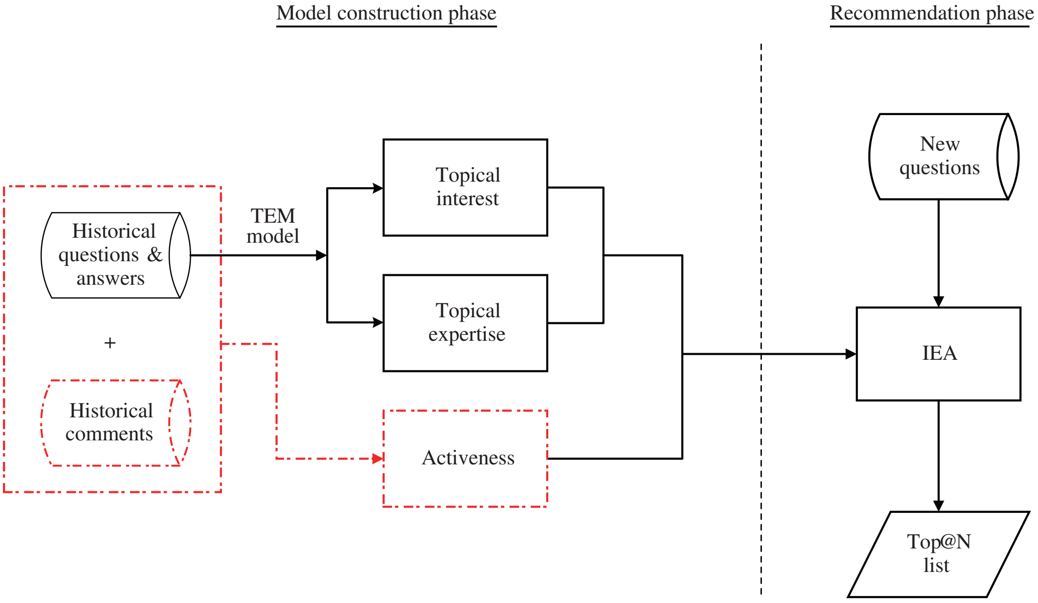


Figure 4 (Color online) Overall framework of our method IEA.

question, we use IEA method to compute users' scores composed by topical interest, topical expertise and activeness, and obtain a rank list of the top N answerers. In Figure 4, top@N is a rank list of the top N answerers who have the highest scores and are recommended. We describe details of recommendation phase in Subsection 3.5.

3.3 Topical interest and topical expertise

Generally, users post questions and answers on stack overflow, and also post comments on questions and answers. Moreover, users vote on questions and answers. The higher the vote scores for questions and answers indicates higher their quality. Vote scores can be used to measure the expertise of users. If users questions or answers always achieve higher votes, users have higher expertise. However, comments posted by users on stack overflow cannot be voted by other users, resulting in no vote scores for comments. Since comments have no vote scores, we are unable to determine the quality of comments. Thus, comments cannot be used to model the expertise of users. For these reasons, comments are not considered in the

Table 3 Symbols associated with TEM

Notation	Type	Description
U	Scalar	The total number of users
N_u	Scalar	The total number of questions and answers for user u
$M_{u,n}$	Scalar	The total number of words in u's n-th question or answer
$L_{u,n}$	Scalar	The total number of tags in u's n-th question or answer
K	Scalar	The total number of topics
E	Scalar	The total number of expertise levels
α	Scalar	Hyperparameter of the Dirichlet prior for the user topic distribution
β	Scalar	Hyperparameter of the Dirichlet prior for the user topical expertise distribution
η	Scalar	Hyperparameter of the Dirichlet prior for the topic-word distribution
γ	Scalar	Hyperparameter of the Dirichlet prior for the topic-tag distribution
α_0, β_0, μ_0, k_0	Scalar	Normal-Gamma parameters
θ_u	Vector	Topic distribution for user u
ϕ_k	Vector	Word distribution for topic k
φ_k	Vector	Tag distribution for topic k
$\theta_{k,u}$	Vector	Expertise distribution for user u under topic k
$G(\mu_e, \Sigma_e)$	Vector	Expertise specific vote distribution

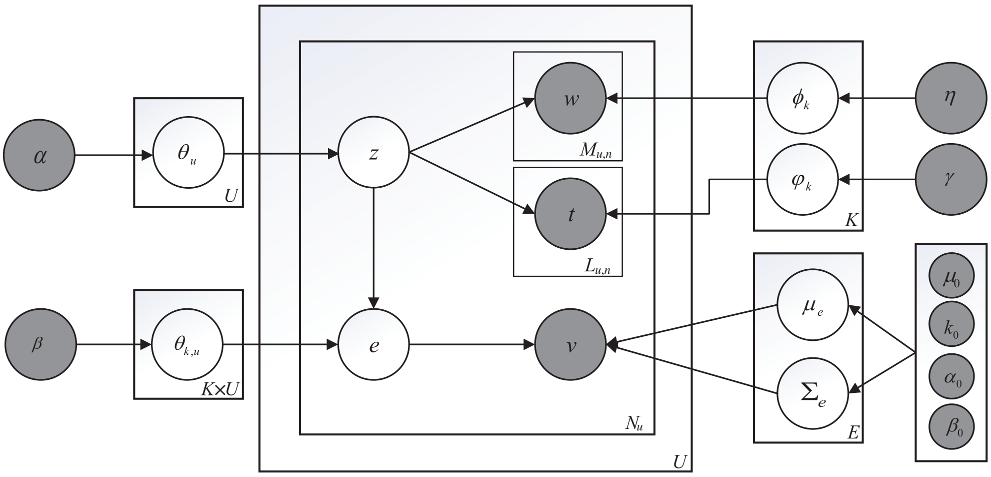


Figure 5 The graphical model of TEM.

computation of user topical interest and expertise for our study.

In order to learn latent topics and expertise of users, we model questions and answers by using TEM proposed in [3] to discover user topical interest and expertise. The TEM is a probabilistic generative model, which jointly models user topical interest and expertise based on historical questions and answers [3]. In the model, the topical interest refers to the user's preference for specific topics on stack overflow. For example, some users prefer content related to "C++", while others are more interested in "Android". The topical expertise refers to user user's expertise level for specific topics on stack overflow. Different users have different topical expertise. In addition, a user may have different expertise levels for different topics.

Table 3 gives the symbols associated with TEM. For clarity, a graphical model of TEM is described in Figure 5. In the figure, the circles represent variables in the graphical model and the rectangles represent variables that repeat a certain number of times. The arrows represent dependencies between variables. The shaded circles are observable variables and known parameters.

In the generative process of TEM, the user topic distribution vector θ_u is firstly selected to determine the probability that each topic of user u is selected. Next, the probability of word w under topic z according to the topic-word distribution vector ϕ_k is generated. In addition, the probability of tag t under topic z according to the topic-tag distribution vector φ_k is obtained. Next, the user topical expertise distribution vector $\theta_{k,u}$ is chosen to determine the probability that each expertise level of user u under topic z is selected. The values of variables θ_u, ϕ_k, φ_k and $\theta_{k,u}$ are calculated by Gibbs sampling [5]. For a given user u, a specific topic k, and an expertise level e, the estimation formula of user topic distribution $\theta_{u,k}$ and user topical expertise distribution $\theta_{k,u,e}$ are shown as

$$\theta_{u,k} = \frac{N_u[k] + \alpha}{\sum_{k=1}^{K} N_u[k] + K\alpha}, \tag{1}$$

$$\theta_{k,u,e} = \frac{N_{k,u}[e] + \beta}{\sum_{e=1}^{E} N_{k,u}[e] + E\beta}, \tag{2}$$

where $N_u[k]$ is the number of questions and answers posted by user u under topic k, $N_{k,u}[e]$ is the number of expertise levels for user u under topic k, K is the total number of topics, and E is the total number of expertise levels.

Finally, the probability of expertise level e of user u under topic z based on the expertise specific vote distribution $G(\mu_e, \Sigma_e)$ is obtained. The estimation formula of (μ_e, Σ_e) is detailed in [3].

$$\mu_e = \frac{k_0\mu_0 + n_e\bar{v}_e}{k_0 + n_e}, \tag{3}$$

$$\Sigma_e = \frac{\alpha_0 + \frac{n_e}{2}}{\beta_0 + \frac{1}{2}\sum_v (v - \bar{v}_e)^2 + \frac{k_0 n_e(\bar{v}_e - \mu_0)^2}{k_0 + n_e}}, \tag{4}$$

where $\bar{v}_e$ is the average vote score for expertise level e, n_e is the total number of votes with expertise level e.

3.3.1 *Topical interest*

A user may answer a specific question if a specific question is relevant to the users interest. New questions fall into some particular topics, and they may be routed to users who are interested in those topics. In order to find users with similar topical preference for new questions, we compute the similarity between interested topics of the user and the topics of the question. Given a new question q, we score each user u by considering the similarity between interested topics of the user and the topics of the question INTEREST(u, q), and the INTEREST(u, q) is defined as

$$\text{INTEREST}(u, q) = 1 - \text{JS}(\theta_{u,k}, \delta_{q,k}), \tag{5}$$

where $\theta_{u,k}$ is the user topic distribution, $\delta_{q,k}$ is the question topic distribution and JS$(\cdot)$ is the Jensen-Shannon divergence distance. The Jensen-Shannon divergence is a method of measuring the similarity between two probability distributions and is bounded by 0 and 1[4].

3.3.2 *Topical expertise*

A user is interested in a topic, but he/she may not be an expert of this topic. Thus, no user is an expert in all interested topics, which means his/her expertise level should be evaluated with respect to the corresponding topics. Different users have different topical expertise, and each user may have different expertise levels for different topics. In order to obtain expertise of each user under different topics, we compute his/her expertise Exp(u, k) for a given user u under a specific topic k. The Exp(u, k) is defined as follows:

$$\text{Exp}(u, k) = \sum_{e=1}^{E} \theta_{k,u,e} \times \mu_e, \tag{6}$$

where E is the total number of expertise levels, $\theta_{k,u,e}$ is the user topical expertise distribution, and μ_e is the probability value of the average vote score for an expertise level e.

[4] Jensen-Shannon divergence. http://en.wikipedia.org/wiki/Jensen-Shannon_divergence.

Besides, in order to find the users with high topical expertise for each new question, we compute his/her expertise EXPERTISE(u,q) for a given user u in a new question q. The EXPERTISE(u,q) is defined as

$$\text{EXPERTISE}(u,q)=\sum_{k=1}^{K}\delta_{q,k}\times\text{Exp}(u,k),\tag{7}$$

where K is the total number of topics, $\delta_{q,k}$ is the question topic distribution and Exp(u,k) is the expertise of a given user u under a specific topic k.

3.4 Activeness

Stack overflow is a CQA website about computer programing. The users on stack overflow always have the freedom to decide their activities and even leave the community. For new questions, some potential answerers may not always be available. Therefore, we need to recommend answerers who are recently active. As described in Figure 3, most active users on a specific day also participate in questions, answers and comments on previous day. Therefore, we study questions, answers and comments to measure users' activeness. We compute his/her activeness ACTIVENESS(u,q) for a given user u in a new question q. The ACTIVENESS(u,q) is defined as

$$\text{ACTIVENESS}(u,q)=\tan\left(\frac{\pi}{2}\times\sum_{1}^{m}(D_q-D_i)^{-2}\right),\tag{8}$$

where m is the number of historical operations for a given user u, D_q is the creation time of the operation for a new question q and D_i is the creation time of the ith operation for a given user u. Historical operations mean questions, answers and comments which are provided by this user. Users who are recently active will have higher activeness.

3.5 IEA

We build an IEA approach by combing user topical interest, topical expertise and activeness to recommend answerers for new questions.

We recommend answerers by combining topical interest, topical expertise and activeness, and give a recommendation list by ranking each answerer. Rec(u,q) is recommendation scores of a given user u for new question q, which is defined as

$$\text{Rec}(u,q)=\text{INTEREST}(u,q)\times\text{EXPERTISE}(u,q)\times\text{ACTIVENESS}(u,q).\tag{9}$$

Given a new question, we compute recommendation scores Rec(u,q) of its candidate answerers. All candidate answerers are ranked according to Rec(u,q), and we can obtain a rank list of answerer recommendation. We select the answerers with the top N highest Rec(u,q) as the recommended answerers for the new question.

4 Experiment results

To evaluate the performance of our proposed approach IEA, we perform some experiments based on the real-world datasets and compare IEA against some existing studies, i.e., TEM [3], TTEA [4] and TTEA-ACT [4].

4.1 Datasets and setup

The stack overflow focuses on the software engineering community, and the real data from stack overflow is used in our experiments. According to Subsection 2.2, we obtain 3428 users from stack overflow as experimental user data. We obtain 40455 questions, 61072 answers and 93054 comments posted by 3428 users from January 1st 2014 to June 30th 2014 as training data. For testing data, we select all questions, answers and comments posted by 3428 users from July 1st 2014 to July 31th 2014. So the training data and the testing data do not have overlap. In testing data, we remove questions which have less than 2 answers according to previous studies [3]. The testing data set contains 1495 questions, 3822 answers and 3906 comments. For data preprocessing, we tokenize text, discard all code snippets and remove the stop words.

We model topics and expertise by using TEM [3]. We set topic number $K = 10$ and expertise number $E = 10$ by default. We discuss settings of topic number and expertise number in Subsections 4.5.4 and 4.5.5.

If a question has several actual answerers, these answerers are ranked by their vote scores. On stack overflow, the questions and answers which are posted by a user have vote scores given by other users. For a question q, the truth rank $T_{q,n}$ of the nth actual answerer is obtained by ranking his/her vote scores. For example, the vote scores of user 1, 2, and 3 are 21, 6, and 13. Therefore, ranks of user 1, 2, and 3 are 1, 3, and 2.

For each question, we recommend answerers by using recommendation scores in (9) from Subsection 3.5. The recommended answerers are ranked by their recommendation scores, and we can obtain the top N answerers of each question. For a question q, the recommendation rank $R_{q,n}$ of the nth recommended answerer is obtained by ranking his/her recommendation scores.

4.2 Baseline methods

In order to evaluate performance of IEA, we describe baseline methods as follows:

- TEM. Previous work [3] proposed a topic expertise model for modeling user topical interest and topical expertise in CQA websites. They modeled user topical interest based on historical questions and answers, and modeled user topical expertise by vote scores. Although they focused on topical interest and topical expertise for users, they did not consider users' activeness. In order to implement TEM method, we downloaded the source code of TEM from GitHub[5]. We read the source code and experiments process of TEM, and then debugged the source code of TEM.
- TTEA, TTEA-ACT. Previous work [4] proposed a TTEA model for modeling user topical interest and expertise in CQA websites. They modeled user topical interest based on historical questions and answers, and modeled user topical expertise by using the logarithmic values of vote scores. They only focused on user topical interest and expertise to recommend answerers, but they did not consider activeness of users. In addition, they computed activity scores of users on new questions to analyze user activeness, and then also proposed TTEA-ACT by combining user topical interest of TTEA and user activeness. However, TTEA and TTEA-ACT did not consider combining user topical interest, topical expertise and activeness to recommend answerers. Furthermore, they only analyzed questions and answers without considering comments. In order to implement TTEA and TTEA-ACT, we implemented the two methods by encoding the description and experimental process of TTEA and TTEA-ACT proposed in previous work [4].

4.3 Evaluation metrics

According to previous studies [4, 6], nDCG, Pearson rank correlation coefficient and Kendall rank correlation coefficient are used to measure performance of answerer recommendation.

The nDCG is used to measure the ranking quality of answerer recommendation. According to previous work [3], the nDCG@N is defined as

$$\text{nDCG@N} = \frac{1}{Q}\sum_{q=1}^{Q}\frac{\sum_{n=1}^{N}(2^{R_{q,n}}-1)/\log_2(n+1)}{\sum_{n=1}^{N}(2^{T_{q,n}}-1)/\log_2(n+1)}, \tag{10}$$

where Q is the number of questions, N is the number of answerers on question q, $R_{q,n}$ is the recommendation rank for the nth answerer of question q, and $T_{q,n}$ is the truth rank for the nth answerer of question q.

[5] https://github.com/yangliuy/TopicExpertiseModel.

According to previous work [3], we also use Pearson and Kendall rank correlation coefficients to measure the strength of correlation between the truth rank list and the recommending rank list. The truth rank list is obtained by ordering truth rank of each answerer, and the recommendation rank list is obtained by ordering recommendation rank of each answerer.

Pearson is the Pearson rank correlation coefficient, which is defined as

$$\text{Pearson} = \frac{1}{Q}\sum_{q=1}^{Q}\frac{N\sum_{n=1}^{N}T_{q,n}R_{q,n} - \sum_{n=1}^{N}T_{q,n}\sum_{i=n}^{N}R_{q,n}}{\sqrt{N\sum_{n=1}^{N}T_{q,n}^2 - (\sum_{n=1}^{N}T_{q,n})^2}\sqrt{N\sum_{n=1}^{N}R_{q,n}^2 - (\sum_{n=1}^{N}R_{q,n})^2}}, \tag{11}$$

where Q is the number of questions, N is the number of answerers on question q, $T_{q,n}$ is the truth rank of the nth answerer on question q, and $R_{q,n}$ is the recommendation rank of the nth answerer on question q.

For the nth answerer of a given question q, $T_{q,n}$ is his/her truth rank and $R_{q,n}$ is his/her recommendation rank. Meanwhile, $T_{q,m}$ is the truth rank of mth answerer of q and $R_{q,m}$ is the recommendation rank of mth answerer of q. If both $T_{q,n} > T_{q,m}$ and $R_{q,n} > R_{q,m}$, or both $T_{q,n} < T_{q,m}$ and $R_{q,n} < R_{q,m}$, the pair is considered as concordant. If ($T_{q,n} > T_{q,m}$ and $R_{q,n} < R_{q,m}$), or ($T_{q,n} < T_{q,m}$ and $R_{q,n} > R_{q,m}$), the pair is considered as discordant.

Kendall is the Kendall rank correlation coefficient, which is defined as

$$\text{Kendall} = \frac{1}{Q}\sum_{q=1}^{Q}\frac{C_q - D_q}{\sqrt{(M-\text{NT})(M-\text{NR})}}, \tag{12}$$

where Q is the number of questions, C_q is the number of concordant pairs for question q, D_q is the number of discordant for question q, $M = \frac{1}{2}N(N-1)$, N is the number of answerers on question q, NT is the number of equal pairs for true ranks, and NR is the number of equal pairs for recommendation ranks. Details are described in previous work[6].

In order to compare two methods, we define the gain to compare how the method 1 outperforms the method 2. As described in initial study [7], nDCG gain, Pearson rank correlation coefficient gain and Kendall rank correlation coefficient gain are defined as

$$\text{Gain}_{\text{nDCG@N}} = \frac{(\text{nDCG@N}(1) - \text{nDCG@N}(2))}{\text{nDCG@N}(2)}, \tag{13}$$

$$\text{Gain}_{\text{Pearson}} = \frac{(\text{Pearson}(1) - \text{Pearson}(2))}{\text{Pearson}(2)}, \tag{14}$$

$$\text{Gain}_{\text{Kendall}} = \frac{(\text{Kendall}(1) - \text{Kendall}(2))}{\text{Kendall}(2)}, \tag{15}$$

where nDCG@N(1), Pearson(1) and Kendall(1) evaluate the performance of method 1, and nDCG@N(2), Pearson(2) and Kendall(2) evaluate the performance of method 2. If the gain value is above 0, it means method 1 has better performance than method 2; otherwise method 2 has better recommendation results.

To further compute two method, we define the ratio to compute how the method 1 outperforms the method 2. The nDCG ratio, Pearson rank correlation coefficient ratio and Kendall rank correlation coefficient ratio are defined as

$$\text{Ratio}_{\text{nDCG@N}} = \frac{M_{\text{nDCG@N\_}(R_{\text{method1}} > R_{\text{method2}})}}{Q}, \tag{16}$$

$$\text{Ratio}_{\text{Pearson}} = \frac{M_{\text{Pearson\_}(R_{\text{method1}} > R_{\text{method2}})}}{Q}, \tag{17}$$

$$\text{Ratio}_{\text{Kendall}} = \frac{M_{\text{Kendall\_}(R_{\text{method1}} > R_{\text{method2}})}}{Q}, \tag{18}$$

[6] Kendall rank correlation coefficient. http://en.wikipedia.org/wiki/Kendall_rank_correlation_coefficient.

where Q is the total number of questions. $M_{\text{nDCG@N\_}(R_{\text{method1}}>R_{\text{method2}})}$ is the number of questions whose nDCG@N value of method 1 is better than the nDCG@N value of method 2. $M_{\text{Pearson\_}(R_{\text{method1}}>R_{\text{method2}})}$ is the number of questions whose Pearson value of method 1 is better than the Pearson value of method 2. $M_{\text{Kendall\_}(R_{\text{method1}}>R_{\text{method2}})}$ is the number of questions whose Kendall value of method 1 is better than the Kendall value of method 2.

In addition, we define the following null hypotheses to assess the statistical validity of results. The alternative hypotheses can be easily derived from the respective null hypotheses.

H-1: There is no statistical significance difference (SSD) between nDCG@1, nDCG@5, nDCG, Pearson, Kendall values of IEA and TEM.

H-2: There is no SSD between nDCG@1, nDCG@5, nDCG, Pearson, Kendall values of IEA and TTEA.

H-3: There is no SSD between nDCG@1, nDCG@5, nDCG, Pearson, Kendall values of IEA and TTEA-ACT.

The Mann-Whitney-Wilcoxon (MWW) test is a non-parametric statistical test that assesses the SSD between two distributions [8]. Therefore, we apply the MWW test to assess SSD of the nDCG@1, nDCG@5, nDCG, Pearson, Kendall values between compared approaches. Test purpose is to assess whether the distribution of one of the two samples is stochastically greater than the other.

4.4 Research questions

In this paper, we are interested in the following research questions:

RQ1: What is performance of IEA as compared with TEM [3], TTEA [4] and TTEA-ACT [4]?

RQ2: What is the benefit of comments in answerer recommendation?

RQ3: What is the benefit of attribution combination in answerer recommendation?

RQ4: What is the effect of varying the number of topics on the performance of IEA?

RQ5: What is the effect of varying the number of expertise on the performance of IEA?

4.5 Results

4.5.1 *RQ1: performance of IEA*

Table 4 shows nDCG@1, nDCG@5, nDCG, Pearson rank correlation coefficient, and Kendall rank correlation coefficient of TEM [3], TTEA [4], TTEA-ACT [4], and IEA. The nDCG@1, nDCG@5, nDCG, Pearson rank correlation coefficient, and Kendall rank correlation coefficient values of IEA are 0.6624, 0.8349, 0.9020, 0.1880, and 0.1649. The gain values of evaluation metrics are described in Table 5.

First, we compare nDCG@1, nDCG@5, nDCG, Pearson rank correlation coefficient, and Kendall rank correlation coefficient values between TEM and IEA. Clearly, IEA outperforms TEM results by 10.29%, 2.68%, 2.48%, 236.20%, and 424.18% in terms of nDCG@1, nDCG@5, nDCG, Pearson rank correlation coefficient, and Kendall rank correlation coefficient. IEA achieves higher nDCG@1, nDCG@5, nDCG, Pearson rank correlation coefficient, and Kendall rank correlation coefficient values, because IEA considers user activeness and comments. In addition, IEA records positive gains with statistical significance (with p-values<0.05) in all cases. Therefore, we find support to reject Hypothesis H-1 in favor of IEA. As shown in Table 5, IEA has better performance than TEM.

Next, compared with TTEA, IEA improves nDCG@1 by 14.53%, improves nDCG@5 by 3.74%, improves nDCG by 3.45%, improves Pearson rank correlation coefficient by 84.91%, and improves Kendall rank correlation coefficient by 1845.30%. It is obvious that IEA outperforms TTEA across nDCG@1, nDCG@5, nDCG, Pearson rank correlation coefficient, and Kendall rank correlation coefficient values. In comparison with TTEA, IEA combines user topical interest, topical expertise and activeness to recommend answerers which improves rank quality of recommended answerers. Furthermore, p-values are smaller than 0.05 in all cases, and IEA records positive gains with statistical significance. Therefore, we find support to reject Hypothesis H-2 in favor of IEA.

Third, in comparison with TTEA-ACT, IEA improves TTEA-ACT by 15.17%, 4.11%, 3.79%, 224.12%, and 772.60% in terms of nDCG@1, nDCG@5, nDCG, Pearson rank correlation coefficient, and Kendall rank correlation coefficient. IEA achieves statistically significant higher nDCG@1, nDCG@5, nDCG,

Table 4 nDCG, Pearson and Kendall of approaches TEM, TTEA, TTEA-ACT, and IEA

	nDCG@1	nDCG@5	nDCG	Pearson	Kendall
IEA	**0.6624**	**0.8349**	**0.9020**	**0.1880**	**0.1649**
TEM	0.6006	0.8131	0.8802	0.0559	0.0315
TTEA	0.5784	0.8048	0.8719	0.1017	0.0085
TTEA-ACT	0.5752	0.8020	0.8690	0.0580	0.0189

Table 5 nDCG gain, Pearson gain and Kendall gain of approaches TEM, TTEA, TTEA-ACT, and IEA

	nDCG@1	nDCG@5	nDCG	Pearson	Kendall
	gain (%)	gain (%)	gain (%)	gain (%)	gain (%)
IEA vs. TEM	10.29 $***$	2.68 $**$	2.48 $**$	236.20 $***$	424.18 $**$
IEA vs. TTEA	14.53 $*$	3.74 $*$	3.45 $*$	84.91 $**$	1845.30 $**$
IEA vs. TTEA-ACT	15.17 $**$	4.11 $**$	3.79 $**$	224.12 $**$	772.60 $**$

$***$ $p < 0.001$, $**$ $p < 0.01$, $*$ $p < 0.05$.

Table 6 nDCG ratio, Pearson ratio and Kendall ratio of approaches TEM, TTEA, TTEA-ACT, and IEA

	nDCG@1	nDCG@5	nDCG	Pearson	Kendall
	ratio (%)	ratio (%)	ratio (%)	ratio (%)	ratio (%)
IEA vs. TEM	89.38	87.19	87.19	82.51	88.05
IEA vs. TTEA	85.63	83.44	83.44	79.88	85.13
IEA vs. TTEA-ACT	86.88	85.31	85.31	79.30	86.59

Pearson rank correlation coefficient and Kendall rank correlation coefficient values than TTEA-ACT in all cases. As shown in Table 5, our proposed IEA outperforms TTEA-ACT. Therefore, we find support to reject Hypothesis H-3 in favor of IEA.

In order to reduce deviation of different questions, the performance is analyzed for each question. More specifically, we determine whether IEA is superior to TEM in term of nDCG@1 for each question. For all questions, we calculate the number of questions whose values of nDCG@1 of IEA are better than TEM, divided by the total number of questions. We also make similar calculations for TTEA and TTEA-ACT in terms of nDCG@5, nDCG, Pearson rank correlation coefficient and Kendall rank correlation coefficient. The ratio values of evaluation metrics are described in Table 6. The experimental results show that in more than 82% of questions, IEA is better than TEM in terms of nDCG@1, nDCG@5, nDCG, Pearson rank correlation coefficient, and Kendall rank correlation coefficient. We also find that in more than 83% of questions, IEA is better than TTEA and TTEA-ACT in terms of nDCG@1, nDCG@5, nDCG and Kendall rank correlation coefficient. For Pearson rank correlation coefficient, IEA is better than TTEA and TTEA-ACT in more than 79% of questions. Therefore, IEA outperforms TEM, TTEA and TTEA-ACT.

IEA achieves worse performance than other methods in a few questions. We manually check some bad cases and try to find reasons. For example, IEA achieves worse performance when some active users change to become inactive. In future work, we will try other functions to measure activeness, and explore performance in recommending answerers.

RQ1. IEA has better performance than TEM, TTEA and TTEA-ACT in terms of nDCG@1, nDCG@5, nDCG, Pearson rank correlation coefficient, and Kendall rank correlation coefficient.

4.5.2 *RQ2: benefit of comments*

In order to answer RQ2, we build another approach IEA-no-comment. This approach does not consider comments to compute activeness. Other parts are the same as IEA.

In Table 7, we compare nDCG@1, nDCG@5, nDCG, Pearson rank correlation coefficient, and Kendall rank correlation coefficient values of IEA-no-comment and IEA. IEA has slightly better performance than IEA-no-comment in terms of nDCG@1, nDCG@5, nDCG and Kendall rank correlation coefficient. For Pearson rank correlation coefficient, IEA is obviously better than IEA-no-comment.

Table 7 nDCG, Pearson and Kendall of approaches IEA-no-comment and IEA

	nDCG@1	nDCG@5	nDCG	Pearson	Kendall
IEA	**0.6624**	**0.8349**	**0.9020**	**0.1880**	**0.1649**
IEA-no-comment	0.6555	0.8328	0.8998	0.1303	0.1602

Table 8 nDCG gain, Pearson gain and Kendall gain of approaches IEA-no-comment and IEA

	nDCG@1	nDCG@5	nDCG	Pearson	Kendall
	gain (%)	gain (%)	gain (%)	gain (%)	gain (%)
IEA vs. IEA-no-comment	1.05	0.26	0.2378	44.30	2.91

Table 9 nDCG ratio, Pearson ratio and Kendall ratio of approaches IEA-no-comment and IEA

	nDCG@1	nDCG@5	nDCG	Pearson	Kendall
	ratio (%)	ratio (%)	ratio (%)	ratio (%)	ratio (%)
IEA vs. IEA-no-comment	95.31	94.06	94.06	93.29	94.46

To investigate the benefit of IEA, the gain values of nDCG@1, nDCG@5, nDCG, Pearson rank correlation coefficient, and Kendall rank correlation coefficient are calculated and results are shown in Table 8. For nDCG@1, nDCG@5 and nDCG, IEA is better than IEA-no-comment. In addition, the proposed IEA method improves Pearson rank correlation coefficient greatly in comparison with IEA-no-comment. The proposed method also outperforms IEA-no-comment in term of Kendall rank correlation coefficient.

In order to reduce deviation of different questions, we compare performance for each question. More specifically, we determine whether IEA is better than IEA-no-comment in term of nDCG@1 for each question. For all questions, we calculate the number of questions whose values of nDCG@1 of IEA are better than IEA-no-comment, divided by the total number of questions. We also make similar calculations for other evaluation metrics. The ratio values of evaluation metrics are described in Table 9. The experimental results show that in more than 95% of questions, IEA is better than IEA-no-comment in term of nDCG@1. We also find that in 94.06% of questions, IEA is better than IEA-no-comment in terms of nDCG@5 and nDCG. For Pearson rank correlation coefficient, IEA is better than IEA-no-comment in 93.29% of questions. For Kendall rank correlation coefficient, IEA is better than IEA-no-comment in 94.46% of questions. Therefore, comments are useful in answerer recommendation.

RQ2. Comments are useful for answerer recommendation. IEA achieves higher nDCG@1, nDCG@5, nDCG, Pearson rank correlation coefficient and Kendall rank correlation coefficient values than IEA-no-comment.

4.5.3 *RQ3: benefit of attribute combination*

In order answer RQ3, the three attributes of topical interest, topical expertise and activeness are combined to evaluate the performance of IEA. In addition to IEA and TEM, five combinations of methods are also used for comparison. The five combinations of methods are topical interest and activeness (TA), topical expertise and activeness (EA), topical interest (INT), topical expertise (EXP), and activeness (ACT), respectively. TEM [3] jointly models topical interest and topical expertise. IEA combines topical interest, topical expertise and activeness to recommend answerers. Table 10 gives the comparison results of IEA with TEM, TA, EA, INT, EXP, and ACT. Besides, the gain values of evaluation metrics are calculated as shown in Table 11.

In Table 10, we compare nDCG@1, Pearson rank correlation coefficient and Kendall rank correlation coefficient values of TEM, TA, EA, INT, EXP, ACT, and IEA. The corresponding gain values of nDCG@1, Pearson rank correlation coefficient and Kendall rank correlation coefficient are described in Table 11. For example, IEA improves TEM by 10.29%, 2.68%, 2.48%, 236.20%, and 424.18% in terms of nDCG@1, nDCG@5, nDCG, Pearson rank correlation coefficient, and Kendall rank correlation coefficient. IEA is better than TA, EA, INT, EXP, and ACT in terms of nDCG@1, nDCG@5, nDCG, Pearson rank correlation coefficient, and Kendall rank correlation coefficient.

Table 10 nDCG, Pearson and Kendall of approaches TEM, TA, EA, INT, EXP, ACT, and IEA

	nDCG@1	nDCG@5	nDCG	Pearson	Kendall
IEA	**0.6624**	**0.8349**	**0.9020**	**0.1880**	**0.1649**
TEM	0.6006	0.8131	0.8802	0.0559	0.0315
TA	0.6333	0.8237	0.8908	0.1180	0.1029
EA	0.6480	0.8297	0.8968	0.1216	0.1497
INT	0.5204	0.7797	0.8467	−0.0685	−0.0908
EXP	0.5586	0.7988	0.8659	−0.0557	−0.0122
ACT	0.6250	0.8205	0.8876	0.0930	0.1063

Table 11 nDCG gain, Pearson gain and Kendall gain of approaches TEM, TA, EA, INT, EXP, ACT, and IEA

	nDCG@1	nDCG@5	nDCG	Pearson	Kendall
	gain (%)	gain (%)	gain (%)	gain (%)	gain (%)
IEA vs. TEM	10.29	2.68	2.48	236.20	424.18
IEA vs. TA	4.60	1.36	1.25	59.40	60.19
IEA vs. EA	2.22	0.63	0.58	54.68	10.13
IEA vs. INT	27.29	7.09	6.53	−374.47	−281.53
IEA vs. EXP	18.58	4.52	4.17	−437.70	−1456.60
IEA vs. ACT	5.99	1.75	1.62	102.19	55.07

Table 12 Performance of IEA by varying the number of topics (T)

	nDCG@1	nDCG@5	nDCG	Pearson	Kendall
$T = 1$	0.6417	0.8274	0.8944	0.1691	0.1310
$T = 2$	0.6458	0.8288	0.8958	0.1394	0.1310
$T = 3$	0.6432	0.8271	0.8942	0.1155	0.1111
$T = 4$	0.6283	0.8218	0.8888	0.1274	0.1012
$T = 5$	0.6420	0.8270	0.8941	0.1586	0.1127
$T = 6$	0.6500	0.8309	0.8979	0.1843	0.1385
$T = 7$	0.6464	0.8289	0.8960	0.1061	0.1277
$T = 8$	0.6343	0.8247	0.8918	0.0886	0.1114
$T = 9$	**0.6633**	0.8347	0.9018	0.1598	0.1583
$T = 10$	0.6624	**0.8349**	**0.9020**	**0.1880**	**0.1649**
$T = 11$	0.6425	0.8271	0.8942	0.1437	0.1332
$T = 12$	0.6231	0.8190	0.8861	0.0901	0.0856
$T = 13$	0.6502	0.8303	0.8973	0.1309	0.1435
$T = 14$	0.6246	0.8213	0.8883	0.1102	0.0869
$T = 15$	0.6242	0.8208	0.8878	0.1277	0.1304

RQ3. The combination of topical interest, topical expertise and activeness is useful for answerer recommendation.

4.5.4 *RQ4: varying the number of topics*

In the subsection, the topic parameter sensitivity is analyzed. We vary the number of topics and observe the change of IEA performance in answerer recommendation. The number of expertise is set as 10 by default.

In the experiment, the topic number T increases from 1 to 15 with an interval of 1. Table 12 demonstrates the performance of IEA with different number of topics. Results show that IEA achieves the highest values of nDCG@5, nDCG, Pearson rank correlation coefficient, and Kendall rank correlation coefficient when the topic number T is set as 10. Therefore, the topic number T is set as 10.

RQ4. IEA achieves the best nDCG@1, nDCG@5, nDCG, Pearson rank correlation coefficient, and Kendall rank correlation coefficient values when the number of topics is set as 10.

Table 13 Performance of IEA by varying the number of expertise (E)

	nDCG@1	nDCG@5	nDCG	Pearson	Kendall
$E = 1$	0.6250	0.8205	0.8876	0.0988	0.1063
$E = 2$	0.6262	0.8202	0.8873	0.1061	0.0760
$E = 3$	0.6257	0.8203	0.8873	0.1009	0.0950
$E = 4$	0.6410	0.8266	0.8937	0.1341	0.1162
$E = 5$	0.6437	0.8282	0.8953	0.1344	0.1087
$E = 6$	0.6309	0.8236	0.8906	0.1146	0.1176
$E = 7$	0.6187	0.8185	0.8855	0.0644	0.0784
$E = 8$	0.6262	0.8218	0.8888	0.1493	0.1161
$E = 9$	0.6070	0.8151	0.8821	0.1204	0.0712
$E = 10$	**0.6624**	**0.8349**	**0.9020**	**0.1880**	**0.1649**
$E = 11$	0.6300	0.8224	0.8895	0.0825	0.0898
$E = 12$	0.6469	0.8301	0.8972	0.1397	0.1511
$E = 13$	0.6328	0.8244	0.8915	0.1111	0.1201
$E = 14$	0.6287	0.8229	0.8900	0.1090	0.1013
$E = 15$	0.6377	0.8246	0.8916	0.1286	0.1210

4.5.5 *RQ5: varying the number of expertise*

In this subsection, we discuss the setting of the number of expertise. Similarly, the expertise parameter sensitivity is analyzed, and the number of topics is set as 10 by default.

The range of expertise number E is from 1 to 15. To investigate effect of the expertise number E, nDCG@1, nDCG@5, nDCG, Pearson rank correlation coefficient, and Kendall rank correlation coefficient values of IEA for different expertise number are calculated as shown in Table 13. Results show that IEA achieves the best performance when the expertise number E is set as 10. Therefore, the expertise number E is set as 10 by default.

RQ5. IEA achieves the best nDCG@1, nDCG@5, nDCG, Pearson rank correlation coefficient, and Kendall rank correlation coefficient values when the number of expertise is set as 10.

5 Threats to validity

Threats to internal validity relate to experimenter bias and errors in our experiments. We consider ground truth as answerers who actually answer questions. We do not know whether actual answerers are the best for making answers of questions. We do not know whether other answerers are equally capable but do not contribute due to issues such as workload and schedule. In this paper, we recommend a rank list of answerers who will really answer new questions, rather than recommending the best answerers. Moreover, our approach is a simple multiplication of topical interest, topical expertise and activeness. In the future, we will consider more ways to combine various factors to improve our approach and achieve better performance.

Threats to external validity relate to generalizability of our study. Our empirical findings are based on open source dataset on stack overflow, and it is unknown whether our results can be generalized to other CQA websites. In the future, we plan to study a similar set of research questions in other CQA websites, and compare their results with our findings on stack overflow.

Threats to construct validity refer to the suitability of our evaluation measures. We use nDCG, Pearson rank correlation coefficient and Kendall rank correlation coefficient, which are also used by previous studies to evaluate effectiveness of answerer recommendation methods [3, 4]. Thus, we believe there is little threat to construct validity.

6 Related work

Related work to this study could be divided into three main categories, including answerer recommendation, study on stack overflow and developer recommendation.

6.1 Answerer recommendation

Current methods for answerer recommendation in CQA websites are mainly based on latent topic modeling techniques, latent expertise modeling techniques and activeness modeling techniques. There have been some previous studies on answerer recommendation [1,3,4,9–12]. Guo et al. [1] proposed a user-question-answer model (UQA) for questions and answerers to discover latent interests of users and recommend question answerers for new arrival questions. Yang et al. [3] proposed TEM for modeling questions and answers to discover experts with both topical interest and expertise, and achieved better performance than UQA [1]. Meng et al. [4] proposed a TTEA model which was used to model topical interest and topical expertise based on historical questions and answers to recommend answerers. Furthermore, they also designed TTEA-ACT model by combing topical interest and user activeness.

Different from these approaches, we propose answerer recommendation approach IEA by combing topical interest, topical expertise and activeness. Experiment results show that IEA achieves higher nDCG@1, nDCG@5, nDCG, Pearson rank correlation coefficient, and Kendall rank correlation coefficient values than TEM [3], TTEA [4] and TTEA-ACT [4].

6.2 Study on stack overflow

There are some previous studies on stack overflow [13–19]. Linares-Vásquez et al. [16] conducted exploratory analysis of mobile development issues on stack overflow. They used topic model to extract main topics which represent mobile development-related discussion on stack overflow. Barua et al. [13] conducted extensive empirical research on all questions and answers of stack overflow. They used latent Dirichlet allocation (LDA) to analyze topics and trends of developer discussions. Beyer and Pinzger [14] manually investigated 450 Android-related questions on stack overflow. They found that the dependencies between question types and problem categories. Nadi et al. [17] performed an empirical investigation into the obstacles developers face with cryptography APIs, through examining questions on stack overflow. Rosen and Shihab [18] narrowed the research scale by focusing on mobile-related questions on stack overflow. They applied LDA based on topic models on the mobile-related questions on stack overflow to investigate topical interest of mobile developers.

Our work is related to but different from above studies. In this paper, our approach combines topical interest, topical expertise and activeness to analyze the recommended answerers.

6.3 Developer recommendation

Recommendation systems dedicated to software engineering can help developers in a wide range of activities. Finding developers is an important requirement in recommendation systems dedicated to software engineering. Some previous studies proposed approaches to assign bug reports or change requests [7, 20–25]. Anvik et al. [20] proposed a semi-automated approach to suggest a small number of developers who resolve bug reports by applying a machine learning algorithm. Jeong et al. [22] proposed a graph model to capture bug tossing history, which is used to assign developer for bug reports. Matter et al. [25] proposed a text-based approach to identify expertise of developers for bug reports. Hossen et al. [21] proposed an approach iMacPro to recommend developers who are most likely to implement incoming change requests. Linares-Vásquez et al. [23] utilized source code authorship to assign expert developers who assist with change request.

Our approach addresses a different problem (answerer recommendation) than those considered in the above mentioned past studies (bug fixer recommendation or change request handler recommendation).

7 Conclusion

In this paper, we first analyze developers' activities by empirical study. Empirical results show that 66.24% of users have more than 30% of comment activities. Comments are important activities for users. In addition, we find that active users in the previous day are likely to be active in the next day. Based on the results of our empirical study, we build a calculation method of users' activeness by historical questions, answers and comments. Then, we use TEM proposed in [3] to model user topical interest and expertise based on historical questions and answers. Therefore, we propose an approach IEA which combines user topical interest, topical expertise and activeness to recommend answerers for new questions. We evaluate the performance of IEA by datasets from stack overflow. The experimental results show that the performance of our approach IEA is better than TEM [3], TTEA [4], and TTEA-ACT [4]. In comparison with TEM, TTEA, TTEA-ACT, IEA improves nDCG@1 by 10.29%, 14.53% and 15.17%, and improves nDCG@5 by 2.68%, 3.74% and 4.11%, and improves nDCG by 2.48%, 3.45% and 3.79%, and improves Pearson rank correlation coefficient by 236.20%, 84.91% and 224.12%, and improves Kendall rank correlation coefficient by 424.18%, 1845.30% and 772.60%.

Acknowledgements This work was supported by National Key Research and Development Program of China (Grant No. 2018YFB1004202), National Natural Science Foundation of China (Grant No. 61672078), and State Key Laboratory of Software Development Environment of China (Grant No. SKLSDE-2018ZX-12).

References

1 Guo J W, Xu S L, Bao S H, et al. Tapping on the potential of Q&A community by recommending answer providers. In: Proceedings of the 17th ACM International Conference on Information and Knowledge Management, California, 2008. 921–930

2 Tian Y, Kochhar P S, Lim E P, et al. Predicting best answerers for new questions: an approach leveraging topic modeling and collaborative voting. In: Proceedings of the 5th International Conference on Social Informatics, Kyoto, 2013. 55–68

3 Yang L, Qiu M H, Gottipati S, et al. Cqarank: jointly model topics and expertise in community question answering. In: Proceedings of the 22nd ACM International Conference on Information and Knowledge Management, San Francisco, 2013. 99–108

4 Meng Z D, Gandon F, Zucker C F. Joint model of topics, expertises, activities and trends for question answering web applications. In: Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence, Omaha, 2016. 296–303

5 Heinrich G. Parameter Estimation for Text Analysis. Technical Report. 2005

6 Järvelin K, Kekäläinen J. Cumulated gain-based evaluation of IR techniques. ACM Trans Inf Syst, 2002, 20: 422–446

7 Xia X, David L, Wang X Y, et al. Accurate developer recommendation for bug resolution. In: Proceedings of the 20th Working Conference on Reverse Engineering, Koblenz, 2013. 72–81

8 Mann H B, Whitney D R. On a test of whether one of two random variables is stochastically larger than the other. Ann Math Statist, 1947, 18: 50–60

9 Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation. J Mach Learn Res, 2003, 3: 993–1022

10 Hu Z T, Yao J J, Cui B. User group oriented temporal dynamics exploration. In: Proceedings of the 28th AAAI Conference on Artificial Intelligence, Québec, 2014. 66–72

11 Wang X R, McCallum A. Topics over time: a non-markov continuous-time model of topical trends. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, 2006. 424–433

12 Zhou G Y, Lai S, Liu K, et al. Topic-sensitive probabilistic model for expert finding in question answer communities. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, Maui, 2012. 1662–1666

13 Barua A, Thomas S W, Hassan A E. What are developers talking about? An analysis of topics and trends in stack overflow. Empir Softw Eng, 2014, 19: 619–654

14 Beyer S, Pinzger M. A manual categorization of Android app development issues on stack overflow. In: Proceedings of the 30th IEEE International Conference on Software Maintenance and Evolution, Victoria, 2014. 531–535

15 Li H W, Xing Z C, Peng X, et al. What help do developers seek, when and how? In: Proceedings of the 20th Working Conference on Reverse Engineering, Koblenz, 2013. 142–151

16 Linares-Vásquez M, Dit B, Poshyvanyk D. An exploratory analysis of mobile development issues using stack overflow. In: Proceedings of the 10th Working Conference on Mining Software Repositories, San Francisco, 2013. 93–96

17 Nadi S, Krüger S, Mezini M, et al. Jumping through hoops: why do java developers struggle with cryptography APIs? In: Proceedings of the 38th International Conference on Software Engineering, Austin, 2016. 935–946

18 Rosen C, Shihab E. What are mobile developers asking about? A large scale study using stack overflow. Empir Softw Eng, 2016, 21: 1192–1223

19 Xu B W, Ye D H, Xing Z C, et al. Predicting semantically linkable knowledge in developer online forums via convolutional neural network. In: Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering, Singapore, 2016. 51–62

20 Anvik J, Hiew L, Murphy G C. Who should fix this bug? In: Proceedings of the 28th International Conference on Software Engineering, Shanghai, 2006. 361–370

21 Hossen M K, Kagdi H, Poshyvanyk D. Amalgamating source code authors, maintainers, and change proneness to triage change requests. In: Proceedings of the 22nd International Conference on Program Comprehension, Hyderabad, 2014. 130–141

22 Jeong G, Kim S, Zimmermann T. Improving bug triage with bug tossing graphs. In: Proceedings of the 7th Joint Meeting of European Software Engineering Conference and ACM SIGSOFT International Symposium on Foundations of Software Engineering, Amsterdam, 2009. 111–120

23 Linares-Vásquez M, Hossen K, Dang H, et al. Triaging incoming change requests: bug or commit history, or code authorship? In: Proceedings of the 28th IEEE International Conference on Software Maintenance, Trento, 2012. 451–460

24 Liu H, Ma Z Y, Shao W Z, et al. Schedule of bad smell detection and resolution: a new way to save effort. IEEE Trans Softw Eng, 2012, 38: 220–235

25 Matter D, Kuhn A, Nierstrasz O. Assigning bug reports using a vocabulary-based expertise model of developers. In: Proceedings of the 6th International Working Conference on Mining Software Repositories, Vancouver, 2009. 131–140