Retrieving People: Identifying Potential Answerers in Brainly

Long T. Le
Rutgers University
longtle@cs.rutgers.edu

Chirag Shah
Rutgers University
chirags@rutgers.edu

Keywords

Community Question-Answering (CQA); Question matching; Expert finding

1. IDENTIFYING POTENTIAL ANSWERERS

In this section, we introduce the problem of finding answerers in CQA, and propose a method to address this challenge.

1.1 Problem definition

The problem is concerned with identifying potential answerers within a CQA community when a new question is posted to that CQA service.

This is an important problem since recommending possible answerers could help in reducing an asker's wait time or increase the likelihood that the question is answered. We propose a framework to predict answerers based on the posting's history as well as features of the question. The first step is building a user profile based on his/her past activities. Then, given a new question q, we compute the score between q and all user profiles. A higher score indicates a higher chance that user will answer the question. Our method includes the following features:

- Similarity between question content and user profile
- Similarity between question topics and user expertise topics
- The active level of user

1.2 Constructing a User Profile

We can build a user profile implicitly or explicitly. Online users might have short descriptions about themselves such as *"Software developer who spent some time in the C++ world but now lives in Eclipse developing java apps"*. From self-declared profiles, we know that this user has expertise in C++, Java and Eclipse. Unfortunately, there are two limitations of building user profiles implicitly. First, many users do not have self-declared profiles or if they do, the description may not be complete. Second, many users do not consistently update their profile with current information. In [3], authors build the user profile based on user opinion such as like/dislikes in online review. A better method is inferring the user profile explicitly. **We can infer the user profile explicitly based on the list of questions they answered. The profile of a user is the concatenation of all the questions they answered.**

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ACM REVIEW '16 USA
Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...$15.00.

1.3 Computing Similarity Between Question and User Profile

Given all user profiles and a new question q, we measure the similarity between q and all user profiles. In order to measure the similarity, we treat each user profile or question as a document. A corpus of documents is built from all user profiles and questions. The next step is computing the *tf-idf* of this corpus [2]. The *tf-idf* value of a word increases linearly with the number of time that the word appears in the document but decreases by the frequency of the word in the corpus. After computing *tf-idf*, each document d is represented by a vector $\vec{\mathbf{v_d}}$. Similarity between a user and a question is measured by the *cosine similarity* between corresponding vectors. The cosine similarity between vectors $\vec{\mathbf{a}}$ and $\vec{\mathbf{b}}$ is described in Equation 2. Cosine similarity is used widely due to its simplicity and efficiency. The cosine similarity in the formula is between 0 and 1 since the *tf-idf* only returns non-negative values. The value of 1 indicates the exact matching while 0 indicates not relevant at all.

$$tf\_idf_{t,d} \;=\; tf_{t,d} \times idf_t \;=\; tf_{t,d} \times log(\frac{N}{df_t}) \tag{1}$$

$$\cos(\vec{\mathbf{a}}, \vec{\mathbf{b}}) = \frac{\vec{\mathbf{a}}\vec{\mathbf{b}}}{\|\vec{\mathbf{a}}\|\|\vec{\mathbf{b}}\|} = \frac{\sum_{i=1}^{n} \mathbf{a}_i \mathbf{b}_i}{\sqrt{\sum_{i=1}^{n} (\mathbf{a}_i)^2}\sqrt{\sum_{i=1}^{n} (\mathbf{b}_i)^2}} \tag{2}$$

1.4 Computing Similarity Between Question Topics and User Expertise Topics

When posting a question in Yahoo! Answers, the user needs to specify the question's category. Some examples of categories in Yahoo! Answers are *poetry, history, painting*. In Stack Overflow, each question has tags such as *java, google-apps-scripts, android-list, jquery*. We can use the categories in Yahoo! Answers or tags in Stack Overflow as question topics. Unfortunately, the category in Brainly is too general in the current system.

Many users do not specify their topics of expertise. However, we can infer the user expertise topics explicitly. The idea is similar to inferring a user profile. Our framework collects the topics of all questions answered by a particular user u. User expertise topics of user u is the concatenation of topics of all questions they answered. If a CQA site does not include information about question topics, we can use topic modeling techniques such as *latent Dirichlet allocation (LDA)* to find the topics of each question [1].

1.5 Activity Level of User

Active users are more likely to answer new questions. Previous work and our work also show that a small fraction of users contribute most of the content in CQA. Users in Yahoo! Answers are awarded points when answering questions. Stack Overflow users also earn reputation value by answering questions. In general, users who give a greater number of answers can earn a higher reputation in Stack Overflow or higher scores in Yahoo! Answers. The majority of CQA sites have some metrics that measure the user activity level. In cases where there is no such metric, we can use the number of posts as the user activity level.

1.6 Finding Potential Answerers

First, we explain how to compute the score between a user and a question. Given a user u and question q, the score between u and q is calculated as in Equation 3. The score is the product between the similarity and the $log(activity\_level)$. The reason for using the log of activity level is the power law distribution of user activity level. In CQA, the activity level of the user follows the power law distribution (shown in Section 2).

$$score(u,q) = log(activity\_level(u)) \times [sim\_contents(u,q) + sim\_topics(u,q)] \quad (3)$$

$$sim(u,q) = sim\_contents(u,q) + sim\_topics(u,q) \quad (4)$$

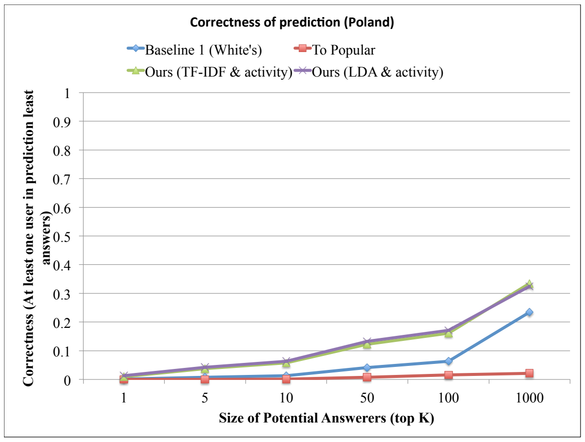
where:

- $activity\_level(u)$ is the activity level of u
- $sim\_contents(u,q)$ is the similarity between question content and user profile
- $sim\_topics(u,q)$ is the similarity between question topics and user expertise topics

Next, we explain our algorithm to find potential answerers. Algorithm 1 describes our *QRec* algorithm. The first step of our algorithm constructs the user profiles. Given a new question q, *QRec* calculates the scores between q and each user. The list of potential answerers are the ones with the top scores. We need to build user profile once (Line 2 in Algorithm 1), and apply for multiple questions. In real application, the user might change his interest. In such case, we can update user profiles but the updating is only needed after a long period (i.e, after a few months). Thus, our ranking method is scalable and can be applied easily on large datasets.

Algorithm 1 *QRec* algorithm

```
Input:
    • A set of $users_i$, $i$ =1, ..., $n$.
    • A question $q$
    • Size of possible answerers $k$
Output: The list of $k$ users most likely answer $q$
1: for $i = 1 : n$ do
2:     Construct the user profile based on self-declared profile and
       answering history
3:     compute scores between $user_i$ and $q$ (as in Equation 3)
4: end for
5: return $topK$: list of $k$ users with top scores
```



(a) Accuracy for Poland market

Figure 1: Compare the correctness by using different approaches to measure the similarity between a user's profile and questions.

2. DATASETS AND CHARACTERIZATION OF THE DATA

2.1 Data Description

We evaluate our algorithm with Poland market. We build the user profile based on the questions he/she answered during Jan 15-Apr 15. We use the question posted in May 15 to test the efficacy of our algorithm.

3. EXPERIMENTS

3.1 Accuracy

Figure 1 compare the accuracy of different algorithm. LDA does not find the topics properly due to short questions in CQA. We strong believe that QRec can achieve higher accuracy if the topics of questions are extracted properly.

4. REFERENCES

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[2] C. D. Manning, P. Raghavan, and H. Schutze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[3] P. Yang and H. Fang. Opinion-based user profile modeling for contextual suggestions. ICTIR '13, pages 80–83, 2013.