International Journal of Engineering & Technology, *7 (3.4) (2018)* 151-159

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Expert Finding in Community Question-Answering for Post Recommendation

Akshi Kumar[1], Saurabh Raj Sangwan[2]*

[1,2]Department of Computer Science & Engineering, Delhi Technological University, Delhi, India
**Corresponding author E-mail:akshikumar@dce.ac.in[1], *saurabhsangwan2610@gmail.com[2]*

Abstract

Community question answering system is a perfect example of platform where people participate to seek expertise on their topic of interest. But information overload, finding the expertise level of users and trustworthy answers remain key challenges within these communities. Moreover, people do not look for personal advices but expert views on such platforms therefore; expert finding is an integral part of these communities. In order to trust someone's opinion who is not known in person by the users of the community, it is necessary to find the credibility of such person. By determining expertise levels of users, authenticity of their posts can easily be determined. Also, by identifying experts, each expert will be shown relevant posts to indulge in so that he can use his knowledge and skills to give valid and correct answers. For users too, it will be easy to find reliable answers, once they get to know the expertise level of the answerers. Motivated by these facts, we put forward a framework for finding experts in online question answer community (stack**overflow**) referred to as Expert Recommender System which uses a well-recognized global-trust metric, PageRank^{TM} for finding experts in the community building a Trust-based system and then uses collaborative filtering to find similar experts based on their level of expertise and their topics of interests to a particular user. Once we have the top- k similar experts to a given expert, that expert is recommended with posts to collaborate upon, based on activities done by his top-k neighbor experts. The framework is evaluated for its performance and it clearly indicates the effectiveness of the system.

Keywords: *Collaborative Filtering; Expert Finder; Global Trust metric; CQA*

1. Introduction

With the advancement of Internet technology and services, information is accessible anywhere, anytime at an inexpensive price. Concurrently, the size of indexed Web and reach of search engines are both increasing at a swift pace. The focus of web-based software has shifted from being typically task-oriented to experience-oriented. UX (or user experience) is the current buzzword which focuses on user engagement and experience and systems which predict the likelihood of content that a user would prefer are ardently needed. Recommender systems or recommendation systems (RS) are one such sub-class of information filtering systems which personalize content. The soul of RS lies in the fact that it should precisely envisage the need and likes of every user [1] but the objectives of RS are dynamic in nature and keep changing as per users' needs. Sometimes, the opinions of similar users to a particular user are helpful but at times the user may not trust a stranger and seek suggestions from a trusted source such as a friend or an expert.

“Ask the Expert” services on Web has been a recent fad where people seek guidance and opinion from certified professionals to resolve financial, career, relationship and health issues. Formally, an expert is a knowledgeable and skillful professional who through theory and practice has the wisdom to give opinion towards a problem or situation. In community-based question & answer (CQA) too, people seek for expert opinions on their problems. Online communities are a great source of information but it is necessary to determine the authenticity of this information in order to spread the knowledge shared on these platforms reliably. Figure 1 enlists the various CQA currently existent on Web.

Fig. 1: Example CQAs

There are many challenges in successful implementation of these communities such as:

- In online communities, the expertise level of the users is not clear and therefore whenever a new user posts any answer or comment, it cannot be determined automatically whether that information can be trusted or not.
- Abundance of information is another significant challenge. Due to overload of information, there is no criterion to make right information available to right set of users. It may happen that a person with high level of expertise in certain area is unable to see all the important posts in that area. So, when a user posts a query it might take excessive amount of time to get accurate and relevant response.
- For every query posted, many people respond whose level of expertise in the given field is not known. So, all right-wrong; true-false answers are accepted without any restriction, creating confusion for the users that which answers can be trusted

These problems are addressed by the expert finding mechanisms used in these communities. By determining expertise levels of users, authenticity of their posts can be easily determined. Also, posts to indulge in so that he can use his knowledge and skills to give valid and correct answers. For users too, it will be easy to find reliable answers, once they get to know the expertise level of the answerers. Expert finding is the process of finding expertise level of each user and identifying erudite people on a given topic. But finding experts in such CQAs is challenging and a dynamic area of research. Usually experts are found via two means - social network analysis (SNA) and concept map. In SNA, individuals are considered as nodes and relationship between them is considered as links of a network. When information is switched between two nodes, a link between them is formed. For example, if person A responds to person B, a link from A is drawn to B. After creating all possible links between all individuals, a network called as Expertise Network (EN) is established [2]. On the other hand, a concept map starts with extraction of concept of user's post and using it to find the expertise level of the user [3]. That is,

- Firstly, each user's posted content is analyzed in order to create a data structure containing concept and keyword related to each post.
- The distance between the concepts is calculated: the concept in a question is mapped to that of an answer associated with it. The output of this stage is a two-dimensional matrix that holds distance between concepts.

In this work, we propose a framework for finding experts in online question answer community (stack**overflow**) using social network analysis. The framework, referred to as Expert Recommender System (ERS) uses a well recognized global-trust metric, PageRank^{TM} for finding experts in the CQAs building a Trust-based ERS and then uses collaborative filtering to find similar experts based on their level of expertise and their topics of interests to a particular user. Pearson Correlation is used as an algorithm for collaborative filtering. That is, the similarity between two users (and their attributes, such as their topics of interest on CQA) can be accurately calculated with the Pearson correlation. This algorithm measures the linear dependence between two variables (or users) as a function of their attributes. The population is filtered down to neighbour-hoods based on a higher-level similarity metric. Thus, once we have the top- k similar experts to a given expert, the expert is recommended [4, 5, 6] with posts to collaborate upon, based on activities done by his top-k neighbour experts. The quintessence of this system lies in the fact that an expert should be recommended with the posts within his domain of interest so that he answers them in a timely manner. Also, if an expert collaborates with other users of same expertise level, then it will make the post, an information rich post and will be beneficial for all the other users of the community. Figure 2 represents the basic premises of the research work undertaken.

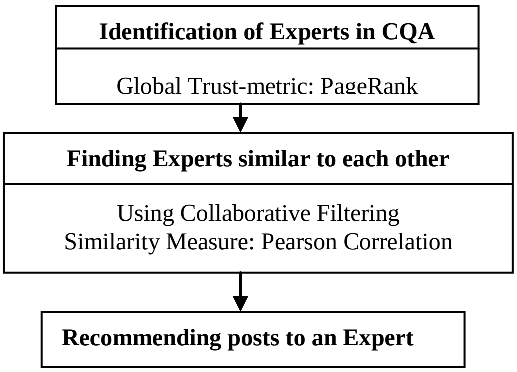

Identification of Experts in CQA
Global Trust-metric: PageRank
Finding Experts similar to each other
Using Collaborative Filtering
Similarity Measure: Pearson Correlation
Recommending posts to an Expert

Fig. 2: The proposed work

The rest of the paper is organized into the following sections. Section 2 gives the background details required for the study which include recommender systems, collaborative filtering, trust-aware recommender system, PageRank, structure of online communities and need of expert finding in online communities and related work in the field. Section 3 describes the proposed framework along with the system architecture. Section 4 expounds the implementa-

by identifying experts, each expert will be shown relevant tion using a sample followed by a discussion on the effectiveness of the framework with the help of the results and analysis in section 5. Finally, section 6 provides the conclusion.

2. Background Work

This section briefs about the pertinent literature discussing details about the recommender system (RS), its types and its applications; Q&A communities (CQA) and the global trust-metric, PageRank. The participative and collaborative social Web has enabled building knowledge structures from the abundant information-base generated by users. But this user-generated content is often varied and voluminous which makes information overload a clear obstacle. That is, the number of options pertaining to products, services and content are overwhelming, and this fosters a need to filter preferred information efficiently alleviating the problem of information overload. Recommender systems (RS) are beneficial in filtering through the big-data offering users personalized content and services. User’s activities, personality, preference, interests and behaviour patterns can be mined to recommend varied things like movies, books, food, clothing, places etc.

Typically, recommender systems are based upon either the information/features provided about a particular object or similarity between two users or items [7]. Based on this, the RS are classified into two broad categories:

- *Content based Recommender Systems (CB):* Content-based systems are based on features of the items recommended. For instance, if an Amazon user has ordered many electronics items, then that user if recommend electronic items classified in the database.
- *Collaborative filtering (CF):* Collaborative filtering finds out similarity between users and items based on their profiles. The items suggested to a user are those that are recommended by similar users.

Collaborative filtering is one of the most commonly used algorithms for recommendation system. Conceptually, it is based on the hypothesis that if we can find some other similar users, what they like might be interesting to you as well [8,9]. However, similarity alone is not an adequate parameter to aid the filtering process. Personalizing the recommendation process can remarkably improve user relevance [10]. The use of social context in the collaborative filtering has been studies across research studies[11]. It primarily involves, adapting a standard collaborateve filtering algorithm to social information, that is, to augment an existing recommendation engine with additional social cues. These cues could be preferences of people in the social network of the user or the people user chooses to follow. This can also enable finding trustworthy and influential people who have expertise in a field. The best example of this concept is asking a friend (who has a similar taste) to recommend a movie you have not seen yet. At the same time we may also turn certified, trustworthy movie critics for reviews. Thus, the approaches for implementing socially-aware recommender systems fall within three categories, namely, the interest-based, the tie-strength based and the trust-based [12]. Figure 3 represents the types of RS.

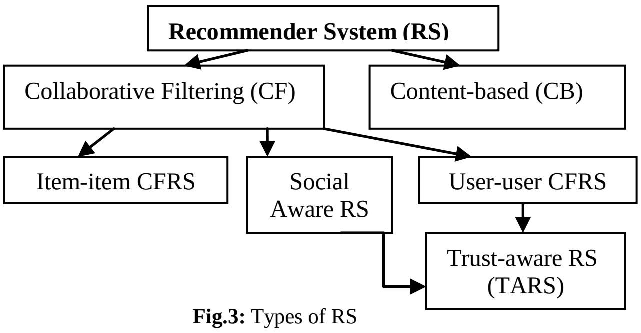


Fig.3: Types of RS

Trust is a relationship of confidence in the ability of others in providing creditable review/rating. It is a concept of behavioural science which creates confidence and decreases uncertainty. Social trust in recommender systems have been widely studies as it can improve the performance of a RS. In 2004, Massa and Avesani [13] proposed a trust-based RS based on a trust metric to predict the trustworthiness of an unknown user by exploiting trust propagation over the trusted network. A comprehensive study of Trust-Aware RS (TARS) has been given by Kumar et.al in 2017[14]. The authors also illustrate the generic structure of TARS and the importance of trust in them. A detailed discussion on the trust metrics is also presented. The authors establish that the strength of any TARS lies in the trust metric it employs. Two categories of trust-metrics have been reported in literature studies, namely, local trust-metrics and global-trust metrics. The key features of both are given in the table 1 below:

Table 1: Difference between Global and Local Trust Metric

	Global Trust-metric	**Local Trust-metric**
Definition	A 'Global' reputation value approximates how the community as a whole consider a user	Takes into account the personal and subjective views of a user and predict different value of trust in other users for every single user.
Key Feature	Trust Propagation: Trust that the entire system places on user i	Personalized: opinion that a user i have on user j, based on past experience.
Example	Google's PageRank	MoleTrust

Online Question-Answering communities (CQAs) are special type of virtual networks where individuals participate for knowledge sharing and seeking. Those who share common interest voluntarily work together so that they can enrich their knowledge by participating in discussions related to topics of their interest. The bow-tie structure of Q&A communities was firstly proposed by researchers at IBM, AltaVista, and Compaq. The key idea behind this was that the web is considered as a bow tie, comprising of four distinct components, i.e. Core, In, Out, and 'Tendrils' and 'Tubes' [15]. This structure of web graph has a central strong connected core (SCC), a sub-graph (IN) with directed path coming into SCC, a component (OUT) leading out of SCC and relatively isolated tendrils attached to one of the three sub-graphs. To frame online community Q&A (CQA) in bow-tie model, it is assumed that the central core contains active users who frequently ask questions and likewise responds to the answers. In a typical real-world scenario, in most communities, the IN component is very large as compared to OUT and SCC, that is, a large number of users utilize these platforms only when they seek help. Moreover, the success and feat of these communities lie in the fact that the questioner should get an answer relevant to his problem in minimum time possible. Further, to choose the right expert to answer a question posted by a user is one of the most challenging problems in the CQAs.

The Google's PageRank^{TM} algorithm, given by Brin & Page [16] in 1998 is a way of deciding a page's importance. To rank the web pages, the importance of a web page is calculated using the value of its neighbouring links. It thus provides a kind of peer assessment by taking into account not just the number of pages linked to it, but also the number of pages pointing to other pages, and so on [15]. PageRank^{TM} has been used widely for finding CQAs. In 2007, Zhang et.al [14] used a set of network-based ranking algorithms, including PageRank and HITS, on large size social network java forum were applied in order to identify users with high expertise and then use these simulations for identifying a small number of simple simulation rules governing the question-answer dynamics in the network. It also proposed a PageRank^{TM} algorithm to find experts in online communities. In 2012, Kardan et.al, also [17] proposed a novel method based on social network analysis for finding the experts in different contexts. Zhao et.al [18], formulated a problem of cold-start expert finding in CQA systems in 2015. Another study by Zhao et.al in 2016 [19] proffered a method to find expert in an online community using both document based relevance and the prestige of the user in his knowledge community. Cheng et.al [20] exploited the user's feedback about the answers provided by a particular user and determined his rank in the community. El-Korany [21] also proposed a novel method to recommend experts to the users of a Q&A online community by considering both content of user and social features. Bozzon et.al, [22] proposed a Competition Based Expertise Networks (CBEN), based on the principle of competition among the answerers of a question. It also showed that the way to determine experts largely depends upon the type of community [23].

To find experts is significant to many real-world applications such as identification of superlative answers and identification of best questions for a user to answer. In online communities the level of knowledge of each user is not known hence it is difficult to decide the quality of an answer. Therefore, by identifying expertise level of users, intelligent systems for knowledge sharing can be built for reliable and credible answers. The next section expounds the details of the proposed framework for finding experts in an online community

3. Proposed Expert Recommender System

This research proffers a novel framework for expert mining in virtual communities. The purpose is two-fold.

- To identify experts in CQAs
- To find similar experts so as to recommend posts within the similar domain of interest of experts.

The recommendation of posts will ultimately create a kind of expert roundup where multiple experts will make the post information-rich. That is, the answers will be more reliable, accurate, and credible. Thus, the proposed Expert Recommender System (ERS) uses a well recognized global-trust metric, PageRank for finding experts in a CQA (stack**overflow**) building a Trust-based ERS and then uses collaborative filtering to find similar experts based on their level of expertise and their topics of interests to a particular user. The main objective of this system is to provide an expert with the posts that may interest him most. If an expert collaborates with other users of same expertise level, then it will make the post, an information-rich post and will be beneficial for all the other users of the community. The general structure of the proposed Expert Recommender system is as follow:

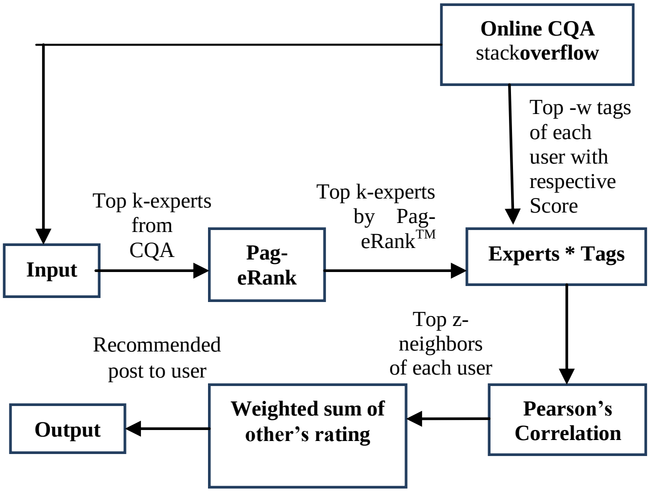
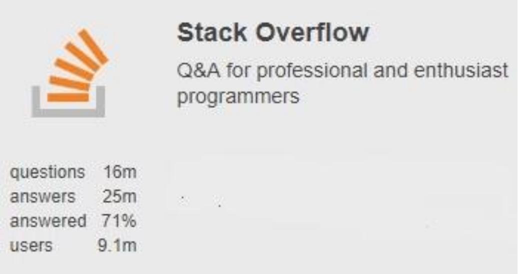

Online CQA
stackoverflow
Top -w tags of each user with respective Score
Top k-experts from CQA
Top k-experts by PagеRank^{TM}
Input
Pag-eRank
Experts * Tags
Top z-neighbors of each user
Recommended post to user
Output
Weighted sum of other's rating
Pearson's Correlation

Fig. 4: Proposed Expert Recommender System

The following sub-sections elaborate the details:

3.1. Data Acquisition

stack**overflow** is a classic example of CQA which as per the recent statistics, currently has 9.1 million users globally. The following figure 5 depicts the social presence of the CQA in terms of number of questions asked, number of answers, percentage of questions answered and the number of users, till date [24].



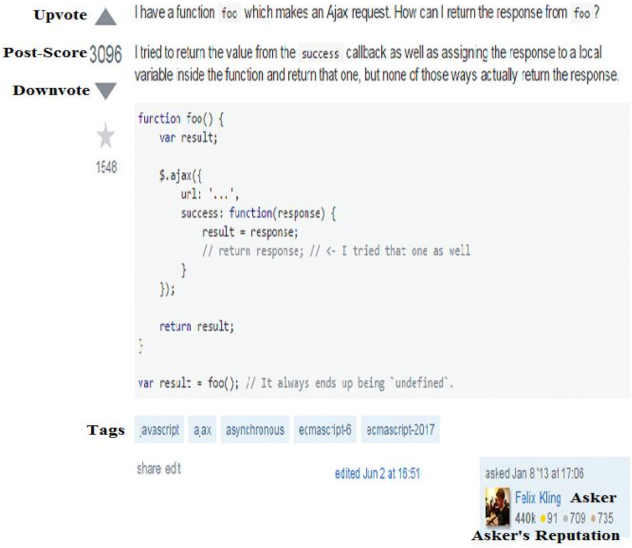
Stack Overflow

Q&A for professional and enthusiast programmers

questions 16m
answers 25m
answered 71%
users 9.1m

Fig.5: Stack Overflow Statistics

stack**overflow** assigns reputation points to each of its users [25]. Reputation is an approximate measure of how much a community trusts a particular user. The primary way to gain reputation is by posting good questions (upvoted question) and useful answers (upvoted answer). So, as an initial step we extract the top k-users along with their reputation score from the CQA **(**stack**overflow**). Next we find the top-z neighbours (on basis of similarity) of each of the extracted top-k experts. A sample post from stack overflow is as shown in figure 6 below:



I have a function foo which makes an Ajax request. How can I return the response from foo?

I tried to return the value from the success callback as well as assigning the response to a local variable inside the function and return that one, but none of those ways actually return the response.

```
function foo() {
    var result;

    $.ajax({
        url: '...',
        success: function(response) {
            result = response;
            // return response; // <- I tried that one as well
        }
    });

    return result;
}

var result = foo(); // It always ends up being `undefined`.
```

javascript ajax asynchronous ecmascript-6 ecmascript-2017

share edit edited Jun 2 at 18:51

asked Jan 8 '13 at 17:06 Felix Kling 440k 91 709 735

Fig.6: Sample stack**overflow** post

3.2. Finding Expert in Stack**overflow**

PageRank™ has been used widely in social network analysis to find experts in an online community. We put forward a technique based on the PageRank™ algorithm to find experts in an online community. The details of the technique are as follows:

Let there be a user X who has helped users $U_1, U_2, ... U_N$. Then the rank (importance) of X can be calculated as follows:

$$PR(X) = (1-d) + d\,[PR(U_1)/Y + PR(U_2)/Y + ... + PR(U_N)/Y] \quad (1)$$

where, with probability d, the page rank of X is the summation of the ranks of all the users whom he has helped divided by the total number of users Y who helped them. Now, the process of finding experts using PageRank™ works as follows:

- Firstly, a list of Top-k users on the basis of their reputation points earned in the community is extracted from their website using their API.
- For each user in Top-k list, top-z posts are extracted on the basis of score of answers provided by the users in those particular posts.
- *for* X, where X is amongst the top-k users
    - *for* each post of X in the top-z list
    - *Extract*
        1. Post score (no. of users who find that the particular post was helpful)
        2. Reputation of user who originally posted the question
        3. Answer count of the posts (no. of answers on that particular question)
- There are 2 parameters which can be used to determine the rank of a particular user 'X' – *post-score* and *user reputation who asked the question to which 'X' has responded*.

  These two parameters are considered to cover the two aspects of reputation which are

  - Total number of people helped through the post *(post-score) →Coverage*
  - How prestigious members of the communities are helped through the posts? *(asker-reputation) →Prestige*

  Therefore, the two ranks will be determined using PageRank™ as follows:

$$PR_{post\text{-}score}(X) = (1-d)+d[PR(Q_1)/Y+PR(Q_2)/Y+.....+PR(Q_N)/Y] \quad (2)$$

Here, PR(Qi) is the score of the i^{th} question (score is the number of people who found this particular question helpful) to which X has provided an answer.
Y is the total number of answers which are accepted for this question.
d is the probability factor used to evaluate PR (d= 0.85 as fixed by [13] as optimum probability)

$$PR_{asker\text{-}reputation}(X) =(1-d)+d[PR(U_1)/Y + PR(U_2)/Y+.....+PR(U_N)/Y] \quad (3)$$

Here, PR(Ui) is the reputation of the i^{th} user to whose question X has responded to.
Y is the total number of answers which are accepted for U_i's question.
d is the probability factor used to evaluate PR (d= 0.85 as fixed by [13] as optimum probability)

Once these two ranks are evaluated for a user 'X', the ranks are normalized on the scale of 0-100. Once both the ranks are on same scale i.e. 0-100, equal weightage of 0.5 is given to each rank and aggregate rank will be computed as:

$$PR_{aggregate}(X) = 0.5 * PR_{post\text{-}score}(X) + 0.5 * PR_{asker\text{-}reputation}(X) \quad (4)$$

Let us consider the following example to understand the ranking mechanism:
Let for user 'X':
Rank based on post-score = 1830
Rank based on asker- reputation = 118670
Normalized rank based on post-score = 87
Normalized rank based on asker-reputation= 54
Aggregate Rank of X will be (0.5*87) + (0.5*54) = 70.5

Thus, we extract data of top- k users of the CQA and then applied our proposed framework on these users and obtained our own list of experts as an output.

3.3. Finding Similar Experts

The expert-neighbourhood is determined using Pearson's correlation. Typically, in a collaborative filtering system, the similarity $w_{u,v}$ between two users u and v, or $w_{i,j}$ between two items i and j, is measured by computing the Pearson correlation (PC). The PC is given as:

$$w_{u,v} = \frac{\sum_{i\in I}(r_{u,i} - \bar{r}_v)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i\in I}(r_{u,i} - \bar{r}_u)^2}\sqrt{\sum_{i\in I}(r_{v,i} - \bar{r}_v)^2}} \quad \textbf{(4)}$$

where i is the total set of items rated by both u and v. $r_{u,i}$ and $r_{v,i}$ are the rating provided for item i by user u and v respectively.
We give variation of this collaborative filtering technique, where

User = Experts determined using page rank
Items = Tags most popularly used in the community.

The resulting matrix will be Experts × Tags and each entry in the matrix will be
$w_{u,v}$ = Score of expert 'u' in tag 'v'

Thus, the parameters of Pearson's Correlation will be defined as follows:
I = set of tags used and each 'i' belongs to I.
u and v = experts
$r_{u,i}$ = score of expert u in tag i
r_u = average score of user u in all the tags rv,i
= score of expert v in tag i
r_v = average score of user v in all the tags

Hence, for an expert 'X', all the top-k neighbours will be identified.

3.4. Recommendation of Posts

For recommendations of posts, the traditional method of Weighted Sum of Others' Ratings is used. To make a suggestion to the active user, a, on a certain item (post), i, a weighted average of all the ratings on that item is taken using the following formula:

$$P_{a,i} = \bar{r}_a + \frac{\sum_{u\in U}(r_{u,i} - \bar{r}_u)\cdot w_{a,u}}{\sum_{u\in U}|w_{a,u}|} \quad \textbf{(5)}$$

where, $P_{a,i}$ = prediction for the active user, a, on a certain item i
U = Set of all the users
r_a and r_u = average ratings for the user a and user u on all other rated items
$w_{a,u}$ = weight between the user a and user u determined by similarity metric used.
The summations are over all the users' u ∈ U who have rated the item i.

Finally, top-k neighbours to each expert will be recorded in the database .Now, Let X is an expert and U is the set of similar experts. Whenever a user who belongs to U, performs an activity that is post a question, post an answer or comment or participate in various ongoing competitions, then user X will be informed about the same and he will be given chances to earn more reputation score by indulging in discussion with people having same level of expertise as him.

the pseudo-code of the proposed system is as follows:

```
Input: Top-k users along with their reputation score from CQA

Output: Top-z neighbours (on basis of similarity) of each of the top-k experts & recommended post to expert

Steps:
1. Extract top-k users from CQA (stackoverflow)
2. Apply PageRank™ to find experts. It includes two measures to compute two scores.
    • Using post-score (Coverage): Post score (no. of users helped) is divided equally among all of them who have answered in that post.
    • Using asker-reputation (Prestige): Reputation of asker is divided equally among all of them who have answered in that post.
3. Finding Aggregate Rank
    • Normalize both the ranks obtained from previous step on a scale of 0-100
    • Giving equal weightage to both coverage and prestige, these two scores are multiplied by 0.5 respectively and then added to obtain aggregate score.
4. Determining the rank for each user on the basis of aggregate rank obtained by them.
5. Extract top-w tags for each expert and produce matrix experts × tags.
6. Complete the matrix with available score of tags for each user from their profiles in the community.
7. Apply Pearson's correlation to obtain top-z most similar experts to a said expert.
8. Using Weighted Sum of Others' Ratings, the user is recommended the posts.
```

4. Implementation

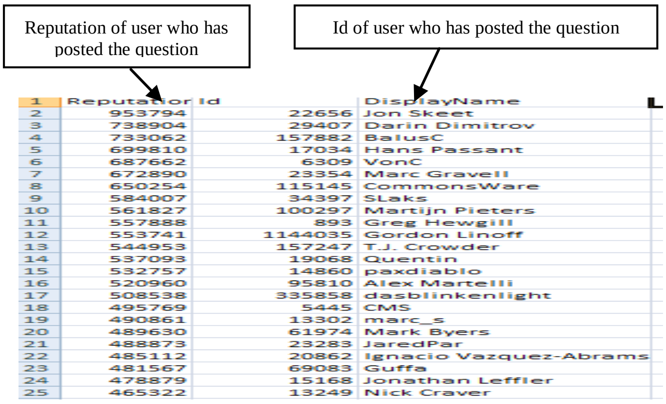
A sample implementation of the proposed expert recommender system on CQA (stack**overflow**) is given in this section. The stack**overflow** API is used to extract required data from the community. The implementation comprises of following steps.

i. *Extraction of Top-50 users of stack overflow on the basis of reputation:* Using SQL query top-50 users of stack overflow are extracted from their database through the API provided by stack exchange which is parent site if stack**overflow**

```
Select TOP(50) Reputation, Id, Displayname from Users Group by Id, Reputation, Displayname der by Reputation desc
```

where, users is the table containing details of all the users, from which we have extracted display name, Id and reputation score of top 50 users of the community. A snapshot of this table containing top 50 users is given in table 2.

Table 2: Snapshot of Top-50 users of Stack overflow


Reputation of user who has posted the question
Id of user who has posted the question

1	Reputation	Id	DisplayName
2	953794	22656	Jon Skeet
3	738904	29407	Darin Dimitrov
4	733062	157882	BalusC
5	699810	17034	Hans Passant
6	687662	6309	VonC
7	672890	23354	Marc Gravell
8	650254	115145	CommonsWare
9	584007	34397	SLaks
10	561827	100297	Martijn Pieters
11	557888	893	Greg Hewgill
12	553741	1144035	Gordon Linoff
13	544953	157247	T.J. Crowder
14	537093	19068	Quentin
15	532757	14860	paxdiablo
16	520960	95810	Alex Martelli
17	508538	335858	dasblinkenlight
18	495769	5445	CMS
19	490861	13302	marc_s
20	489630	61974	Mark Byers
21	488873	23283	JaredPar
22	485112	20862	Ignacio Vazquez-Abrams
23	481567	69083	Guffa
24	478879	15168	Jonathan Leffler
25	465322	13249	Nick Craver

ii. *For each user in top-50 list, top-20 posts are extracted where user has answered in the particular post.* Posts are ordered by votes that the user has achieved on an answer provided by him in a post. For example: User A has post P1 as top post where his answer has got 500 votes and post P2 achieved 2nd position with user A's answer on post P2 has achieved 480 votes and so on. Stack overflow's database contains a table known as 'posts' which contains details of all the posts of the community.

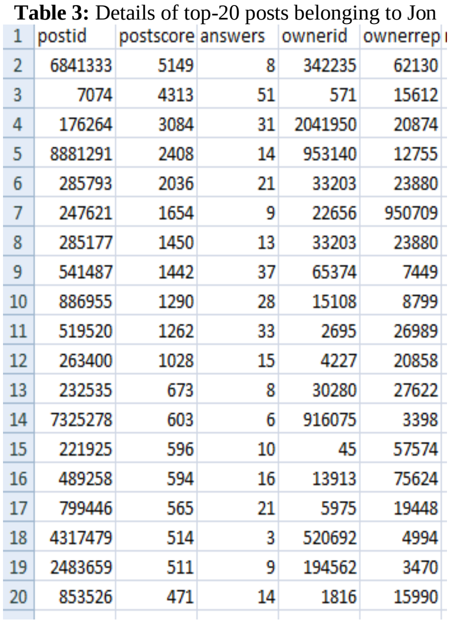
We have extracted post Id, post score, reputation of asker, and answer count for each post. For example, Jon skeet is the top user of stack overflow. To extract top 20 posts of Jon, where he has answered in the post, his Id is used, which is extracted in the previous table.

User name: Jon Skeet

Id: 22656

```
select postid as [Post Link],postscore,users.id as ownerid,
users.reputation as ownerreputation
FROM
(select posts.id as postid, posts.score as postscore, posts.owneruserid
as owner
FROM
(SELECT TOP(21) Id as [Post Link], posts.owneruserid, parentId as
par, score
from Posts where posts.owneruserid = 22656 order by score desc) as
jon JOIN Posts
ON posts.id = jon.par) as jon1 JOIN Users
ON Users.id = jon1.owner
order by postscore desc
```

Table 3: Details of top-20 posts belonging to Jon



1	postid	postscore	answers	ownerid	ownerrep
2	6841333	5149	8	342235	62130
3	7074	4313	51	571	15612
4	176264	3084	31	2041950	20874
5	8881291	2408	14	953140	12755
6	285793	2036	21	33203	23880
7	247621	1654	9	22656	950709
8	285177	1450	13	33203	23880
9	541487	1442	37	65374	7449
10	886955	1290	28	15108	8799
11	519520	1262	33	2695	26989
12	263400	1028	15	4227	20858
13	232535	673	8	30280	27622
14	7325278	603	6	916075	3398
15	221925	596	10	45	57574
16	489258	594	16	13913	75624
17	799446	565	21	5975	19448
18	4317479	514	3	520692	4994
19	2483659	511	9	194562	3470
20	853526	471	14	1816	15990

Similarly, Details of Top-20 posts of all the top-50 users of stack**overflow** are extracted.

iii. Next, for each user two ranks will be calculated using the PageRank^{TM} algorithm in which: One is based on the post-score and another on the reputation of the asker.

- On the basis of asker-reputation: Let there be a user ‘A’, who has asked a question and he has a reputation score of 5000. If a user's question is answered by 50 users, then according to PageRank^{TM}, his reputation is equally divided into all the 50 users irrespective of the content provided by each user. Let a user ‘X’ answer A's question, then contribution of ‘A’ to the rank of ‘X’ will be

  Reputation of A/Total no. of answers to A's question
  Here, contribution of ‘A’ to the rank of ‘X’ will be
  5000/50 = 100 points.

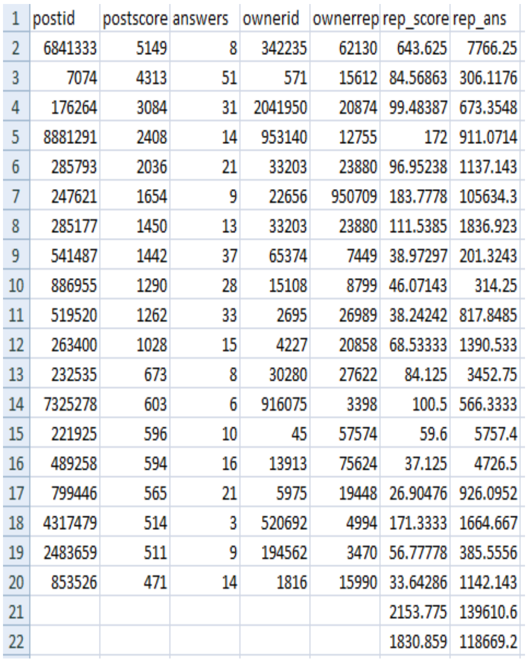
- On the basis of post-score : A user ‘A’ answers a question Q and 1020 users find post useful whereas 20 users find it useless, then the post score(upvotes - downvotes) will be 1000. According to PageRank^{TM}, if a question is answered by 50 users, then its score is equally divided into all the 50 users irrespective of the content provided by each user. A user ‘X’ who answered a question Q, then contribution of Q to the rank of ‘X’ will be

  Post score of Q / Total no. of answers to question Q

  Here, contribution of Q to the rank of X will be
  1000/50 = 20 points.

Similarly, for a particular user, reputation earned by him in all the top 20 posts will be calculated using equations (2) and (3).
For example: User name: Jon Skeet
Here,
rep_score = postscore / answers
rep_ans = ownerrep / answers
2153.775 is the summation of all the rep_score
139610.6 is the summation of all the rep_ans
Rank on basis of post score = 0.15 + (0.85 * 2153.775) = 1830.859
Rank on basis of post score = 0.15 + (0.85 * 139610.6) = 118669.2

Table 4: Details of top-20 posts belonging to Jon along with rep_score and rep_ans



1	postid	postscore	answers	ownerid	ownerrep	rep_score	rep_ans
2	6841333	5149	8	342235	62130	643.625	7766.25
3	7074	4313	51	571	15612	84.56863	306.1176
4	176264	3084	31	2041950	20874	99.48387	673.3548
5	8881291	2408	14	953140	12755	172	911.0714
6	285793	2036	21	33203	23880	96.95238	1137.143
7	247621	1654	9	22656	950709	183.7778	105634.3
8	285177	1450	13	33203	23880	111.5385	1836.923
9	541487	1442	37	65374	7449	38.97297	201.3243
10	886955	1290	28	15108	8799	46.07143	314.25
11	519520	1262	33	2695	26989	38.24242	817.8485
12	263400	1028	15	4227	20858	68.53333	1390.533
13	232535	673	8	30280	27622	84.125	3452.75
14	7325278	603	6	916075	3398	100.5	566.3333
15	221925	596	10	45	57574	59.6	5757.4
16	489258	594	16	13913	75624	37.125	4726.5
17	799446	565	21	5975	19448	26.90476	926.0952
18	4317479	514	3	520692	4994	171.3333	1664.667
19	2483659	511	9	194562	3470	56.77778	385.5556
20	853526	471	14	1816	15990	33.64286	1142.143
21						2153.775	139610.6
22						1830.859	118669.2

Now to bring these two ranks to the same scale, they are normalized to a scale of 0-100 by considering minimum value of post score = 0
minimum value of owner reputation = 0
maximum value of post score = 2100
maximum value of owner reputation = 220000

Thus, the score of Jon Skeet
Normalized post-score= 87
Normalized asker-reputation = 54

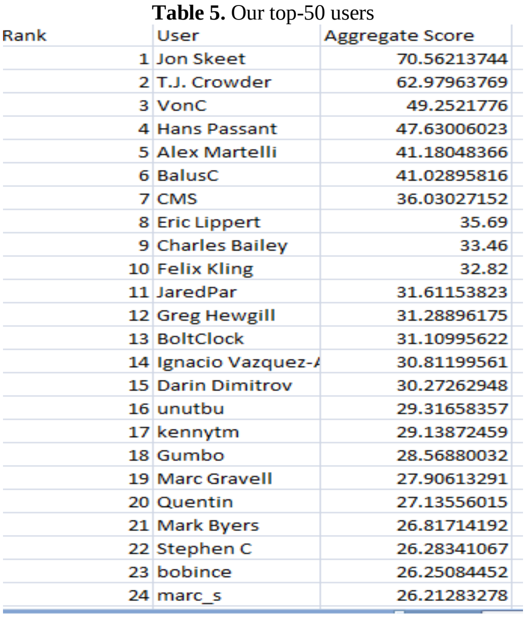
Aggregate rank of Jon (giving equal weightage to both the scores)

= (0.5 * 87) + (0.5*54) = 70.56

Similarly, aggregate scores for all the top-50 users are calculated.
iv. *Ranking of all the users according to our calculated aggregat-*

ed scores: Top-50 users are determined by sorting the obtained aggregate score of all the users in descending order. The table 5 containing the snapshot of our top-50 users is shown below:

Table 5. Our top-50 users

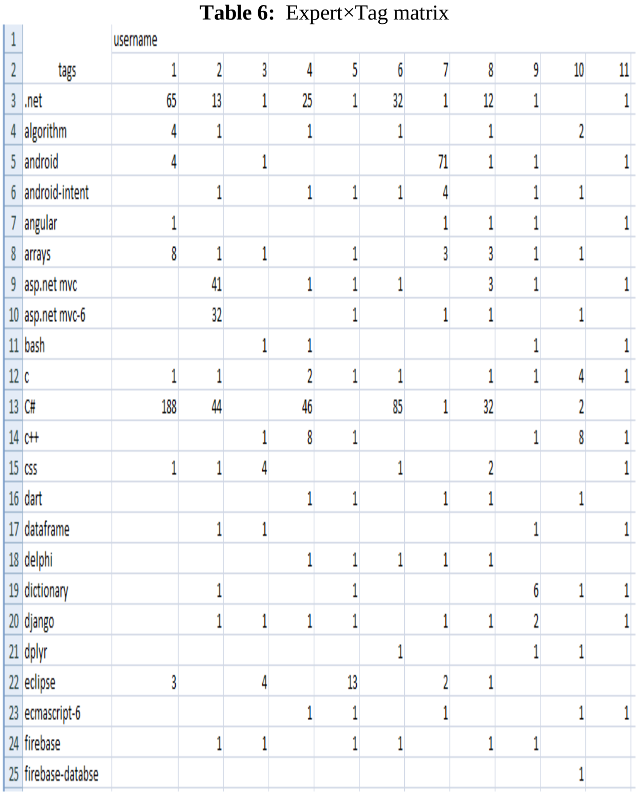


Rank	User	Aggregate Score
1	Jon Skeet	70.56213744
2	T.J. Crowder	62.97963769
3	VonC	49.2521776
4	Hans Passant	47.63006023
5	Alex Martelli	41.18048366
6	BalusC	41.02895816
7	CMS	36.03027152
8	Eric Lippert	35.69
9	Charles Bailey	33.46
10	Felix Kling	32.82
11	JaredPar	31.61153823
12	Greg Hewgill	31.28896175
13	BoltClock	31.10995622
14	Ignacio Vazquez-A	30.81199561
15	Darin Dimitrov	30.27262948
16	unutbu	29.31658357
17	kennytm	29.13872459
18	Gumbo	28.56880032
19	Marc Gravell	27.90613291
20	Quentin	27.13556015
21	Mark Byers	26.81714192
22	Stephen C	26.28341067
23	bobince	26.25084452
24	marc_s	26.21283278

v. *Extraction of Top-3 tags of each user in Top-k list along with their scores for the respective tags and construction of Expert×Tag matrix.* Due to non-availability of features in API of stack**overflow** which can relate each user to its top tags, tags and their respective scores have been extracted manually. To avoid the cumbersome task of extraction of data, top-30 users are selected to put in the Expert×Tag matrix and only top-3 tags of each user are extracted to depict the process of our proposed framework. Once top-3 tags for each user are obtained, the matrix is completed by obtaining the score of all the tags that are in the list (top-3 tags of all the users) for each user.

The sample of the matrix is shown in the following table 6:

Table 6: Expert×Tag matrix



1		username										
2	tags	1	2	3	4	5	6	7	8	9	10	11
3	.net	65	13	1	25	1	32	1	12	1		1
4	algorithm	4	1		1		1		1		2	
5	android	4		1				71	1	1		1
6	android-intent		1		1	1	1	4		1	1	
7	angular	1						1	1	1		1
8	arrays	8	1	1		1		3	3	1	1	
9	asp.net mvc		41		1	1	1		3	1		1
10	asp.net mvc-6		32			1		1	1		1	
11	bash			1	1					1		1
12	c	1	1		2	1	1		1	1	4	1
13	C#	188	44		46		85	1	32		2	
14	c++			1	8	1				1	8	1
15	css	1	1	4			1		2			1
16	dart				1	1		1	1		1	
17	dataframe		1	1						1		1
18	delphi				1	1	1	1	1			
19	dictionary		1			1				6	1	1
20	django		1	1	1	1		1	1	2		1
21	dplyr						1			1	1	
22	eclipse	3		4		13		2	1			
23	ecmascript-6				1	1		1			1	1
24	firebase		1	1		1	1		1	1		
25	firebase-databse										1	

Here all the values are in thousands i.e. score of user 1 for .net is 65k.

Now all the values are normalized by putting a score of 1-5 for the intervals as follow:

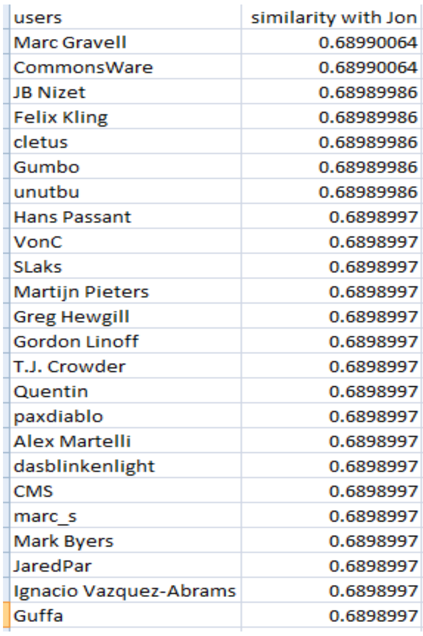
1-25 →	1
26-50 →	2
51-75 →	3
76-100 →	4
>100 →	5

vi. *Applying pearson's correlation for each user, to find his correlation with all the other users in the list:* Pearson correlation is used to find similarity between two given users. For each user in top-30, his correlation is determined with all the other users in the list.

For example:
user name: Jon Skeet

In the output of pearson's correlation, a table (table 7) is produced which contains all the users that are similar to Jon skeet in descending order of their similarity measure.

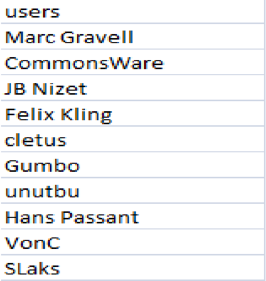
Table 7: users that are similar to Jon skeet



users	similarity with Jon
Marc Gravell	0.68990064
CommonsWare	0.68990064
JB Nizet	0.68989986
Felix Kling	0.68989986
cletus	0.68989986
Gumbo	0.68989986
unutbu	0.68989986
Hans Passant	0.6898997
VonC	0.6898997
SLaks	0.6898997
Martijn Pieters	0.6898997
Greg Hewgill	0.6898997
Gordon Linoff	0.6898997
T.J. Crowder	0.6898997
Quentin	0.6898997
paxdiablo	0.6898997
Alex Martelli	0.6898997
dasblinkenlight	0.6898997
CMS	0.6898997
marc_s	0.6898997
Mark Byers	0.6898997
JaredPar	0.6898997
Ignacio Vazquez-Abrams	0.6898997
Guffa	0.6898997

vii. *Finding Top-10 neighbors of a particular User:* When correlation of a user 'A' with all the other users is arranged in descending order. Top-10 experts in this list will be neighbors of 'A'. Select top-10 users from the above result table who, when indulged in any activity, Jon skeet will be notified about the same.Top-10 neighbors of Jon are in table 8 as below:

Table 8: Top-10 neighbors of Jon



users
Marc Gravell
CommonsWare
JB Nizet
Felix Kling
cletus
Gumbo
unutbu
Hans Passant
VonC
SLaks

The next section illustrates the results and its analysis.

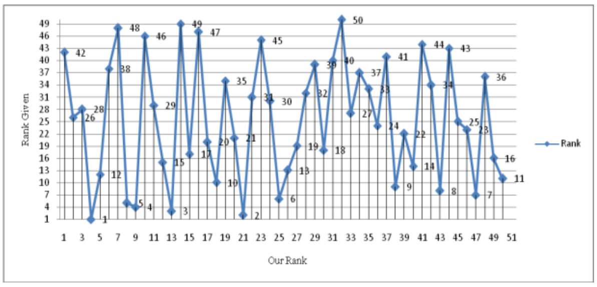
5. Results & Analysis

To study the effectiveness of the proposed system, Top-50 users according to Global Trust metric (PageRank^{TM}) and its correlation with the list of top-50 users from the community is calculated using spearman's rho which is a measure of the degree of agreement between two rankings. It is calculated with and without outliers in the data. Spearman's Rho is a product–moment correlation coefficient devised as a measure of the degree of agreement between two rankings.

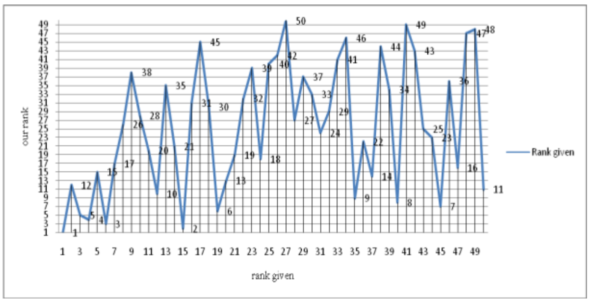
$$r_s = \left[1 - \frac{6\sum D^2}{N^3 - N} \right]$$

where, D, is the difference between the two ranks of each observation. n is the number of observations,
n is the number of observations.

- *Correlation with Outliers in data*
  To calculate correlation between the two ranks, spearman's correlation is used.
  Correlation (rank given, our Rank) = 0.003794



- *Correlation Without Outliers in data*
  Correlation (rank given, our Rank) = 0.50



From the results obtained, it is observed that there are outliers in the data which are affecting our results. For example, if a user X has answered questions of other users with an average reputation of 100 but in one post he has answered a user whose reputation is 10000, then this entry will be considered as an outlier as it will affect the aggregate score of user X. Due to presence of outliers in entries of every user, the correlation between the two ranks (one provided by stack**overflow** and the other that is determined by using PageRank^{TM}) is extremely low.

After calculating the correlation with the presence of outliers, we have calculated the correlation by removing the extreme outliers, which improved the results. As this is the limitation of social network analysis that it divides the reputation of a user among all of his helpers, irrespective of the contribution made by each helper. This limitation has also affected our results. For example, if a post has score of 5000 and it is answered by 100 users but only top 10 users has answered very well as compared to rest of the users, so it will be unfair to divide the post score equally in all the users who have contributed in the particular post. Due to rules of social network analysis, we are restricted to divide the post score equally. This also affects the results and reduced the correlation between two lists of top- 50 (one provided by stack**overflow** and other that is determined by using PageRank^{TM}) users.

Using Pearson's correlation, correlation of each user A is determined with every other user and then out of them, top-10 neighbours will be selected that are like user A. Whenever a user who is neighbour to a user "A" performs an activity that is post a question, post an answer or comment or participate in various on-going competitions, he will be informed about the same and will be given chances to earn more reputation score by indulging in discussion with people having same level of expertise as him.

6. Conclusions & Future Work

People seek suggestions from trusted experts for their questions on online communities. The utility of this system lies in the fact that these communities face the problem of information abundance and to get right person involved in the right threads of questions and answers is one of the biggest challenge. The proposed system makes sure that an expert is provided with relevant questions of his fields of interest and due to contribution of people of same level of expertise and same kind of interests, the threads will be information rich which is beneficial for all the users of the community. We have taken a very small dataset for the sake of simplicity, but results are clearly showing the utility of the proposed work. Our next step involves testing the framework on a larger dataset and on a real-time system. Few are the limitations of the work includes the following:

- Outliers: We have observed that there are outliers in the data which are affecting our results to a large extent. For eg. if a user X has answered questions of other users with an average reputation of 100 but in one post he has answered a user whose reputation is 10000, then this particular entry will be considered as an outlier as it will affect the aggregate score of user X.
  Due to presence of outliers in entries of every user, the correlation between the two ranks is affecting adversely.

- Equal weightage assigned to all the participating users: This is the limitation of social network analysis that it divides the reputation of a particular user among all of his helpers, irrespective of the contribution made by each helper. This limitation has also affected our results. For eg. if a post has score of 5000 and it is answered by 100 users but only top 10 users has answered very well as compared to rest of the users, so it will be unfair to divide the post score equally in all the users who have contributed in the post.

- Reputation is not at all depending on the concept involved in Q&A: It does not include the similarity between the concept involved in the question and that in answer and the reputation is purely determined by the asker's reputation and post score.

As a future direction, following measures can be incorporated. A weightage can be assigned to the similarity measure between the concept involved in questions and answers. It starts with extraction of concept of user's post and using it to find the expertise level of the user. Weightage should be assigned to each answerer on the basis of votes he has achieved for his answer on a particular post and reputation of asker or post score should be divided among the helpers on the basis of basis of weightage given to each of them. Also, tag-specific recommendation can be included where a user is recommended with post in which his neighbours are indulging and also that post should involve user's top tags. This will make the recommender system more precise and each user's expertise will be utilized optimally and at the same time, the user will be provided with opportunities to enhance his/her knowledge within the topics of interest.

References

[1] P. Resnick & H. R. Varian, Recommender systems. Communications of the ACM, 40(3), (1997), 56-58

[2] J. Zhang, MS. Ackerman & L. Adamic, Expertise networks in online communities: structure and algorithms. In Proceedings of the 16th international conference on World Wide Web, ACM, (2007), 221-230.

[3] M. Rafiei & AA Kardan, A novel method for expert finding in online communities based on concept map and PageRank. Human-centric computing and information sciences, *5*(1), (2015), 10.

[4] A. Kumar & N. Ahmad, Comex miner: Expert mining in virtual communities. International Journal of Advanced Computer Science and Applications (IJACSA), (2012), *3*(6).

[5] A. Kumar & A. Sharma, Alleviating sparsity and scalability issues in collaborative filtering based recommender systems. In Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA), Springer, (2013), 103-112.

[6] A. Kumar, MPS Bhatia, Community expert based recommendation for solving first rater problem. International Journal of Computer Applications, 37(10), (2012), 7-13.

[7] F. Ricci, L. Rokach & B. Shapira, Introduction to recommender systems handbook. In Recommender Systems Handbook (pp. 1-35). Springer US (2011).

[8] X. Su & TM. Khoshgoftaar, A survey of collaborative filtering techniques. Advances in artificial intelligence, (2009), 4.

[9] JL. Herlocker, JA. Konstan & J. Riedl, Explaining collaborative filtering recommendations. In Proceedings of the 2000 ACM conference on Computer supported cooperative work, ACM, (2000), 241-250.

[10] Y. Shi, M. Larson & A. Hanjalic, Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. ACM Computing Surveys (CSUR), 47(1), (2014) 3.

[11] P. Lops, M. De Gemmis & G. Semeraro, Content-based recommender systems: State of the art and trends. In Recommender systems handbook (pp. 73-105). (2011), Springer, Boston, MA.

[12]E. Gilbert and K. Karahalios, Predicting tie strength with social media. In Proc. CHI, (2009), 211–220.

[13] P. Massa & P. Avesani, Trust-aware recommender systems. In Proceedings of the 2007 ACM conference on Recommender systems, ACM, (2007), 17-24.

[14] A. Kumar, InduBala & S. Jain, A comprehensive study of TARS: Definition, metrics and advancements. In Electrical, Computer and Electronics (UPCON), IEEE, (2017), 199-204.

[15] S. Chakrabarti, Mining the Web: Discovering knowledge from hypertext data. Elsevier, (2002).

[16] L. Page, S. Brin, R. Motwani & T. Winograd, The PageRank citation ranking: Bringing order to the web. Stanford InfoLab (1999)

[17] A. Kardan, A. Omidvar & M. Behzadi, Context based expert finding in online communities using social network analysis. International J of Computer Science Research and Application, 2(1), (2012), 79-88.

[18] Z. Zhao, F. Wei, M. Zhou & W. Ng, Cold-start expert finding in community question answering via graph regularization. In International Conference on Database Systems for Advanced Applications, Springer, (2015), 21-38.

[19] Z. Zhao, Q. Yang, D. Cai, X. He & Y. Zhuang, Expert Finding for Community-Based Question Answering via Ranking Metric Network Learning. In *IJCAI*, (2016), 3000-3006.

[20] X. Cheng, S. Zhu, G. Chen & S. Su, Exploiting user feedback for expert finding in community question answering. In Data Mining Workshop (ICDMW), 2015 IEEE International Conference, IEEE, (2015), 295-302.

[21] A. El-Korany, Integrated expert recommendation model for online communities. arXiv preprint arXiv:1311.3394, (2013).

[22] Ç. Aslay, N. O'Hare, LM. Aiello & A. Jaimes, Competition-based networks for expert finding. In Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval, ACM, (2013), 1033-1036.

[23] A. Bozzon, M. Brambilla, S. Ceri, M. Silvestri & G. Vesci, Choosing the right crowd: expert finding in social networks. In Proceedings of the 16th International Conference on Extending Database Technology, ACM, (2013), 637-648

[24] "StackOverflow Documentation,"[online],Available: https://stackoverflow.com/documentation/documentation/topics

[25] "StackOverflow API,"[online],Available: https://data.stackexchange.com/stackoverflow