Exploiting Content Quality and Question Difficulty in CQA Reputation Systems

Adrian Huna, Ivan Srba, and Maria Bielikova

Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Ilkovicova 2, 842 16 Bratislava, Slovakia
{xhunaa,ivan.srba,maria.bielikova}@stuba.sk

Abstract. Community Question Answering (CQA) systems (e.g. StackOverflow) have gained popularity in the last years. With the increasing community size and amount of user generated content, a task of expert identification arose. To tackle this problem, various reputation mechanisms exist, however, they estimate user reputation especially according to overall user activity, while the quality of contributions is considered only secondary. As the result, reputation usually does not reflect the real value of users' contributions and, moreover, some users (so called reputation collectors) purposefully abuse reputation systems to achieve a high reputation score. We propose a novel reputation mechanism that focuses primarily on the quality and difficulty of users' contributions. Calculated reputation was compared with four baseline methods including the reputation schema employed in Stack Exchange platform. The experimental results showed a higher precision achieved by our approach, and confirmed an important role of contribution quality and difficulty in estimation of user reputation.

Keywords: Community Question Answering, User Reputation, Expertise estimation

1 Introduction

The Internet is an enormous source of information which helps lots of people every day. Despite the amount of information available, there are still situations in which it is difficult to find specific information, or to answer a question that is too complex to be understood by a search engine. These types of situations led to creation of online communities whose members are focused on helping each other in a specific area. In the past years, especially many Community Question Answering (CQA) systems have appeared and gained popularity among users. They are essentially based on social interactions through asking and answering questions. In addition, all members of CQA communities can vote on the provided answers with the aim to select the most useful one among them. Moreover, the asker can pick any answer and mark it as the best answer, what also serves as an expression of its quality. All questions and answers are publicly available,

and thus CQA systems serve as valuable centers of community knowledge. In general, we can distinguish two types of CQA systems: universal systems consisting of categories from physics, to love or psychology (e.g. Yahoo! Answers); and specialized systems, which focus only on a specific area (e.g. StackOverflow that concerns with programming).

Users in CQA systems exhibit different kinds of behavior and thus create various internal structures of their communities. A traditional problem in systems that employ user generated knowledge is how to simply distinguish authoritative and expert users, who have a great impact on the evolution of the community, from newcomers or less experienced users. Most CQA systems include some kind of method to calculate user reputation as a way to rank users. Identification of high-reputation users is important in order to extend their rights in managing the community, to mentor them for better engagement with the site, or to route hard questions. In addition, visualization of reputation in the user interface allows users to easily recognize users' overall expertise.

These reputation mechanisms, however, often employ very simple principles based primarily on the amount of user activity in the system (regardless the real quality and difficulty of carried out contributions), what leads to an inaccurate reflection of user expertise and their overall value for the community. Moreover, these reputation mechanisms can be very easily abused by so called reputation collectors. There are many sources of data in CQA systems that can be analyzed in order to calculate users' reputation more accurately. It is possible to observe users' behavior in terms of asking and answering questions, look at feedback provided by a community, or study a social graph between askers and answerers. We suppose, that especially by utilization of the community-perceived quality and estimated difficulty of users' contributions, we will be able to measure user reputation more precisely than reputation schemas currently employed in the CQA systems or than methods proposed in the previous works.

2 Related Work

In the current CQA literature, problem of expert identification is commonly based on estimation of various user-related measures, such as:

1. user topical expertise (also termed as a user knowledge profile [4]),
2. user authority, and
3. user reputation.

These measures and their denominations are often used interchangeably and thus the differences between them are commonly neglected. In this paper, we distinguish between these terms as follows:

1. *Differences in Meaning*: In general, the common characteristic of all three measures is that they are indicators of user expertise and capture an amount of user knowledge and his/her potential to provide high-quality answers. User reputation as well as user authority refers to a *global* value of the user to the

community that depends on quality of his/her contributions and activity in the system. In other words, the more expert answers a user can provide, and the more frequently he/she participates in the question answering process, the more authority and reputation he/she should have. On the other side, user topical expertise relates to a *particular topic* (i.e. a user assigned tag/category or an automatically extracted topic).

2. *Differences in Representation*: Both user authority and user reputation are usually represented by a *single value* that provides simple comprehensive information about the user and thus it can be easily displayed in the user interface or utilized to rank users. On the other side, user topical expertise is rather a *more complex variable* that naturally depends on particular topics. It can be used in situations when identification of experts on a certain topic is important, for example in recommendation of recently posted questions to potential answerers (so called question routing).
3. *Differences in Calculation*: We can broadly divide the existing methods to expert identification into graph-based and feature-based approaches. The graph-based approaches work with a social graph underlying users' interactions in CQA systems (mainly between askers and answerers). Various graph-based algorithms (e.g. algorithms developed to rank websites, such as PageRank and HITS) are then applied on these graphs in order to identify authoritative and expert users in the community. The second group of feature-based approaches is based on historical question-answering records about users as well as about content created by them. Consequently, various mostly numerical methods are employed to derive user expertise.
   User authority methods belong to graph-based approaches as they are based on *link analyses*. On the contrary, user reputation methods can be characterized as feature-based approaches – reputation can be calculated either by *reputations schemas* (rule-based mechanisms commonly employed in the existing CQA systems) or *numerically derived* from users' question answering history. Finally, user topical expertise methods can employ either graph-based or feature-based approaches, however, with data limited only to particular topics.

2.1 Reputation Schemas in the Existing CQA Systems

In spite of the large body of research publications on CQA systems, just few of them tackle explicitly with their reputation schemas. The most popular CQA systems utilize user reputation as a part of their gamification systems in order to provide users with motivation to actively participate on question answering.

Users in CQA system Yahoo! Answers are divided into 8 categories based on their reputation score. Each level has limitations in a number of questions and answers a user can contribute each day. Users gain and lose reputation based on their actions in the system. The reputation schema of CQA systems in Stack Exchange platform also work on point based reputation rules[1]. The actions and corresponding reputation changes are displayed in Tab. 1.

[1] http://stackoverflow.com/help/whats-reputation

Table 1. Reputation rules in Stack Exchange platform

Action	Reputation change
Answer is voted up	+10
Question is voted up	+5
Answer is accepted	+15 (+2 to acceptor)
Question is voted down	-2
Answer is voted down	-2 (-1 to voter)
Experienced Stack Exchange user	onetime +100
Accepted answer to bounty	+bounty
Offer bounty on question	-bounty

Analyses of Stack Exchange reputation schema and its influence on user behavior has been performed by Bosu et al. [1] and Movshovitz-Attias et al. [6]. Bosu et al. [1] focused on exploring the ways how users earn reputation in StackOverflow community. They provide an analysis of variations in community's activity between different topics as well as throughout different days during a week and hours during a day. The results of this analysis consist of recommendations how users in StackOverflow can build reputation quickly and efficiently, such as by answering questions related to tags with lower expertise density, answering questions promptly or being active during off peak hours. Differently, Movshovitz-Attias et al. [6] analyzed behavior of users with both high and low reputation. The results showed that high reputation users provide the majority of all answers. On the other hand, the majority of all questions is asked by low reputation users, nevertheless high reputation users ask in average more questions as low reputation ones. Authors also demonstrated the application of their results in a prediction whether a user will become an influential long-term contributor by consideration of contributions in the first months of his/her activity in the system.

Paul et al. [7] studied reputation and its influence on user behavior in CQA system Quora. Quora does not employ any kind of public reputation schema or a visual representation of user reputation, however, there is available another implicit measure of reputation by means of number of user's followers. The lack of reputation system is also compensated by users' individual feeling of satisfaction as well as competency.

Reputation schemas employed in the existing popular CQA systems are based on simple rules in order to be transparent for a community. In addition, system administrators can simply influence the community behavior by gamification in order to promote insufficient actions in the system (e.g. by giving them more reputation points).

2.2 Measuring User Authority and Reputation

Besides rule-based reputation schemas applied in the existing popular CQA systems, it is possible to find several more or less simple measures of user expertise in the research papers concerned with CQA systems. In the following review,

we focus primarily on methods aimed to estimate user authority and user reputation as their common goal is to calculate the global value of users while they differ only in the employed calculation approach.

An early attempt in expert identification in CQA systems was made by Jurczyk et al. [3] who compared performance of two graph-based approaches on different types of graphs and with data from different categories in Yahoo! Answers, particularly HITS algorithm and a simple degree measure (a difference between a number of ingoing and outgoing connections in the question answering graph). The results revealed that HITS algorithm achieved substantially unbalanced performance, it worked well in some categories, while in others its performance was quite week.

Zhang et al. [11] studied users' expertise in a system called Java Forum. Authors proposed the graph-based algorithm named ExpertiseRank, which is inspired by PageRank. However, the biggest influence for further research in this area comes from their proposal of a new feature-based reputation measure called Z-score. It is based only on a number of answers and questions a user contributed:

$$Z_{\text{score}} = \frac{a - q}{\sqrt{a + q}} \tag{1}$$

where a represents a number of posted answers and q is a number of asked questions. The authors also provided a comparison between graph-based and feature-based approaches, in which a simple Z-score metric performs better than other graph-based methods.

Liu et al. [5] proposed another graph-based approach that utilizes pairwise competition, i.e. the relationship between the best answerer and other answerers supposing that the best answerer has a higher expertise as other answerers. In comparison with the previous graph-based approaches, algorithms for ranking players (e.g. TrueSkill) were employed. The effectiveness of these ranking methods was compared with traditional graph-based algorithms (PageRank and HITS) and also with simple feature-based approaches (number of answers, number of best answers, best answer ratio and smoothed best answer ratio). The results showed that the proposed competition-based approach achieved very similar performance as much simpler feature-based metric best answer ratio.

2.3 Influence of Activity on User Expertise Estimation

Yang et al. [10] pointed out a problem that is present in standard expert identification methods. These methods very often misclassify very active users (denoted by authors as sparrows) for experts (denoted as owls). While sparrows generate most of the content, owls provide valuable answers to questions that are perceived as important by the community. The existing expert identification methods, however, targeted mainly sparrows as they focused mainly on the amount of users' activity in the system rather than on quality of their contributions. As the result, methods for topical expertise, authority as well as reputation estimation suffer with a serious issue - the calculated estimation of user expertise does not usually reflect real users' knowledge level.

The similar problem is present also in reputation schemas employed in the existing CQA systems. The negative consequences of these reputation schemas, which also favor user activity, lie in reputation abuse. As we showed in our previous case study [9] aimed to analyze user behavior in StackOverflow, we can observe increasing population of reputation collectors and other kinds of undesired types of users. Reputation collectors intentionally abuse the reputation system in order to collect reputation by answering as many questions as possible (commonly regardless their insufficient knowledge on the particular question topic).

To address these drawbacks, it is necessary to propose novel methods that balance the influence of user activity and quality of contributions. At first, Yang et al. [10] focused on the quality of users' contributions for topical expertise estimation. Authors proposed a metric called Mean Expertise Contribution which takes question debatableness and answer utility into calculation in order to distinguish sparrows and owls more precisely.

Instead of contribution quality, question difficulty was taken into consideration by Hanrahan et al. [2] in order to identify expert users more precisely. Authors decided to use duration between the time when the question was asked and the time when an answer was marked as the best answer as the measure for question difficulty. Authors, however, did not propose any method for reputation estimation, only observed correlation between question difficulty and user expertise represented by StackOverflow reputation and Z-score.

The conclusions from the analyzed state-of-the-art approaches to user expertise estimation provide directions for a proposal of our method. At first, feature-based approaches not only perform better than graph-based ones but also are computationally more efficient. Secondly, in feature-based approaches, it is essential to distinguish between user activity and quality of contributions. In spite of that, the most of existing approaches give a priority on the amount of user activity. An exception is the method by Yang et al. [10] that addressed this issue in estimation of *user topical expertise*. On the other side, we are not aware of any similar solution proposed for user reputation estimation.

3 Calculating User Reputation with Content Quality and Difficulty

Our main goal is to model users' reputation with accentuation on the quality of users' contributions, not their activity as it is done in the reputation schemas employed in the popular CQA systems and in the existing feature-based methods, in order to estimate user reputation with better success rate.

In our approach, reputation of a user consists of reputation gained for:

1. providing answers on questions asked by the rest of the community, as well as for
2. asking new questions.

It is in the contrast to methods for user topical expertise estimation (e.g. [10]) that usually consider only providing answers. The reason is that answering a question can be perceived as an expression of expertise on question topics, while asking a question, on the other side, can be perceived as a lack of expertise. However, in estimation of user reputation, asking popular questions as well as providing good answers is important.

A user gains greater reputation for asking difficult and useful questions and for providing useful answers on other difficult questions. The gained reputation for such actions is added to previously earned reputation. Final reputation R of a user u can thus be expressed as a sum of reputations gained for asking questions R_{q}, summed up with a sum of reputations gained for answering questions R_{a}. Formula (2) represents the formal expression of the final reputation:

$$R(u) = \sum R_{\mathrm{q}}(q) + \sum R_{\mathrm{a}}(a, q) \tag{2}$$

We also propose an alternative formula in order to completely suppress an influence of an amount of users' activity:

$$R(u) = \frac{\sum R_{\mathrm{q}}(q) + \sum R_{\mathrm{a}}(a, q)}{|q| + |a|} \tag{3}$$

where $|q|$ is the number of questions a user asked and $|a|$ is the number of answers he/she provided.

3.1 Reputation for Asking Questions

Inspired by the work [2], we propose to calculate reputation for asking questions based on question difficulty D_{q} in a combination with question utility QU. We suppose that the longer it takes for the first answer to be added (time to answer a question q - $TTA(q)$), the more difficult the question is. In order to take into account differences between various topics in CQA systems, we normalize this time by maximum time to add the first answer for questions assigned to the same topic t ($TTA_{\max}(t)$). If a question belongs to more topics, we calculate D_{q} for each topic, and then average the results. We decided to use a logarithm of TTA values in order to solve a long tail distribution of the values. The binary logarithm is used because it performed better than the natural and the common (decadic) logarithm. Question difficulty D_{q} for a question q is computed as:

$$D_{\mathrm{q}}(q) = \frac{\log_2 (TTA\,(q))}{\log_2 (TTA_{\max}\,(t))} \tag{4}$$

The second factor for calculating reputation for asking questions is question utility QU. Our formula for question utility is an adaptation of an idea in the work [10]. We calculate question utility as $Score$ (number of positive votes minus number of negative votes) normalized by a maximum value of scores - $MaxScore(t)$ on questions in the same topic t to reflect differences in popularity between topics in CQA systems. If a question belongs to more than one topic,

we calculate QU for every topic, and then we average the results. In addition similarly as for question difficulty, a logarithm of scores is used because we can observe a long tail distribution also for questions' scores.

$$QU(q) = \frac{\log_2 (Score\,(q))}{\log_2 (MaxScore\,(t))} \tag{5}$$

In the calculation, we had to solve several specific situations. At first, if a question receives negative score, question utility will be negative too. To calculate negative utility more accurately, we use absolute value of minimum question score for a topic t in the place of $MaxScore(t)$. Secondly, if a score of a question is zero and $MaxScore(t)$ is zero as well, QU will be equal one. Finally, we adapted the logarithm calculation in order to be able to handle negative values and zero. The logarithm of negative values is calculated as $-\log_2(-x)$ and the logarithm of zero is zero.

The final form of formula for reputation obtained for asking questions consists of sum of question difficulty and question utility. Formula (6) displays the final relationship for calculating reputation R_{q} for asking a question q:

$$R_{\mathrm{q}}(q) = D_{\mathrm{q}}(q) + QU(q) \tag{6}$$

3.2 Reputation for Answering Questions

The second part of our reputation system, which is responsible for calculating reputation for answering questions, utilizes question difficulty (4) as described in the previous section, and combines it with answer utility which adapts an idea from the work [10]. Answer utility $AU(a, q)$ for an answer a in a question q is calculated as:

$$AU(a, q) = \frac{\log_2 (Score\,(a))}{\log_2 (MaxAnswerScore\,(q))} \tag{7}$$

where $Score(a)$ is a score of an answer a, and $MaxAnswerScore(q)$ represents a maximum score from all answers provided for a question q. If an answer receives a negative score, answer utility will be negative too, as the same approach as for question utility is used. If $Score$ and $MaxAnswerScore$ are both equal zero, and the answer is labelled as *the best* then answer utility is equal one, otherwise zero. The best answer status, however, has no effect on answer utility for answers with nonzero score. The reason for using logarithm of answers' scores is the same as for logarithm of questions' scores with the same rules for negative values.

As well as in (6), we use the sum of question difficulty and answer utility for calculating reputation gained for answering a question:

$$R_{\mathrm{a}}(a, q) = D_{\mathrm{q}}\,(q) + AU\,(a) \tag{8}$$

4 Evaluation

4.1 Experiment Setup

In order to evaluate the proposed reputation system, we conducted an offline experiment in which we used two datasets from CQA systems Programmers[2] (collected in September 2014) and Sharepoint[3] (collected in August 2015), which are parts of Stack Exchange network. The data are publicly available to download on archive.org[4].

We are not aware of any gold standard available for the Stack Exchange datasets that could be used to evaluate the calculated users' reputations against. At first, there is not such a thing as an absolute value representing real user reputation since all existing scoring metrics are calculated according to a certain heuristic method that itself can be considered as an approach to estimate user reputation. In addition, datasets do not contain a global list of all users in the community sorted relatively according to their reputation either. Utilization of human judgements is not applicable here because it is not possible to manually evaluate so many users and all their previous activities in the system [11].

As the result of missing gold standard, many alternative approaches have been already employed in the previous works. The most objective way to evaluate the performance of user reputation estimation without manual data labelling, which is not applicable on large datasets, is a utilization of partial rankings of users. More specifically, it is possible to compare two sorted lists of users for each question separately. The first list is sorted according to calculated reputation, while the second one is sorted according to the score of answers as accumulated in the CQA system (if two answers have the same score, we consider the newer as better one assuming that the previous one did not answer the question sufficiently). This gives us the ability to evaluate how many users are in their correct position as well as examine the difference in rankings between these two lists.

As a baseline for comparison, we chose four feature-based approaches:

1. Firstly, we have reconstructed the original user reputation based on Stack Exchange reputation rules.
2. As the second method for comparison, we chose Best Answer Ratio (BAR) for each user, which performed as the best in the previous works.
3. As the third method, we chose Z-score, as proposed by Zhang et al. [11].
4. Finally, we employed a number of previously posted answers, which reflects only user activity and totally ignores quality of provided contributions.

As our method works with question difficulty, which is based on time to answer a question, we can take into consideration only those questions that have at least one answer. Moreover, we evaluated the performance of all methods for only those questions which have at least two answerers with calculated reputation, so

[2] http://programmers.stackexchange.com/
[3] http://sharepoint.stackexchange.com/
[4] https://archive.org/details/stackexchange

we could perform a comparison between the lists of users (users with unknown reputation were left out from the comparison). For these reasons, we report our results on about 20 000 questions even though there are 33 052 questions in the Programmers dataset, and on about 11 000 questions from total number of 47 136 questions in the Sharepoint dataset respectively.

The evaluation was performed employing an experimental infrastructure, a part of CQA system Askalot [8] which is being developed at Faculty of Informatics and Information Technologies at Slovak University of Technology in Bratislava. The infrastructure enables us to reconstruct events as they happened in time, thus allows us to perform the chronological evaluation process.

4.2 Evaluation Metrics

Standard information retrieval metrics are applied in order to compare the performance of our method and baselines:

- Precision at N (P@N): The proportion of top N users who are ranked at the correct position.

$$P@N = \frac{r}{N} \tag{9}$$

  where r is the number of users in the correct position.
- Mean Reciprocal Rank (MRR): The reciprocal rank is the inverse of position (according to the ground truth) for the user with highest reputation (evaluated by the proposed method). The mean reciprocal rank is the average of reciprocal ranks for all questions evaluated:

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i} \tag{10}$$

  where $|Q|$ is the number of questions, and $rank_i$ is the position of the user.
- Normalized Discounted Cumulative Gain (nDCG): A method which uses graded relevance as a measure of usefulness. Positions of users in the beginning of the list are more important than positions in the end of the list. The formula stands as follows:

$$nDCG = \frac{DCG_p}{IDCG_p} \tag{11}$$

  where DCG_p is Discounted Cumulative Gain, and $IDCG_p$ is the ideal possible DCG - it is DCG of the ground truth, while DCG_p is Discounted Cumulative Gain of users sorted according a method being evaluated. We use alternative formulation of DCG:

$$DCG_p = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2 (i + 1)} \tag{12}$$

  where p is a rank position evaluated, rel_i is relevance of a user at a position i.

4.3 Evaluation Results

In order to evaluate how individual components of the proposed method for reputation calculation contribute to user reputation, we evaluated its performance in two steps. Firstly, we worked only with reputation gained for answering questions (labeled as *Answers only*). Secondly, we employed also reputation for asking questions (i.e. the full variant of the proposed method). We also examined two configurations of our method in order to completely eliminate activity factor (Formula (3) labeled as *average*), and Formula (2) labeled as *sum* in the results.

Table 2 reports the results of our experiments on the Programmers dataset and Tab. 3 on the Sharepoint dataset, respectively. We present performance of Precision@1 (P@1), Precision@2 (P@2), Mean Reciprocal Rank (MRR) and Normalized Discounted Cumulative Gain (nDCG). The last column displays the number of questions which were evaluated.

Table 2. Comparison of the performance of the methods on the Programmers dataset

	P@1 (%)	P@2 (%)	MRR (%)	nDCG (%)	Questions
Full variant *(sum)*	40.093	38.074	65.538	83.162	20552
Full variant *(average)*	**43.971**	**41.154**	**66.278**	84.511	20552
Answers only *(sum)*	40.179	38.267	63.632	83.233	20324
Answers only *(average)*	43.623	40.926	66.182	**84.521**	20324
Stack Exchange Reputation	**42.080**	39.279	**64.850**	**83.888**	20558
Best Answer Ratio	41.881	**40.078**	64.585	83.728	20324
Z-score	38.388	37.022	62.322	82.534	20558
Number of answers	38.570	37.308	62.481	82.647	20324

Table 3. Comparison of the performance of the methods on the Sharepoint dataset

	P@1 (%)	P@2 (%)	MRR (%)	nDCG (%)	Questions
Full variant *(sum)*	36.005	37.324	65.554	85.429	11451
Full variant *(average)*	**50.004**	**49.563**	**73.145**	**88.671**	11451
Answers only *(sum)*	35.754	37.017	65.450	85.410	11042
Answers only *(average)*	45.634	45.707	70.753	87.692	11042
Stack Exchange Reputation	34.895	36.397	64.904	85.168	11483
Best Answer Ratio	**40.481**	**41.441**	**67.870**	**86.425**	11042
Z-score	35.313	36.693	65.177	85.309	11483
Number of answers	35.020	36.424	65.016	85.235	11042

The results show that our method outperformed all baseline methods. The interesting observation is that the variant which completely eliminates user activity performed as the best. This result confirms the significant influence of the quality of user contributions. It is especially true for the Sharepoint dataset, for which the methods that emphasize user activity perform clearly worse than the ones that suppress it (i.e. best answer ratio and our method in average variant).

In addition, we observe differences also between the full and partial (i.e. answers only) variant of our method. The full variant reflects user reputation

better because it captures reputation gained from answering as well as asking questions. While on the Programmers dataset the differences are not so obvious, the full variant outperforms the answers only variant by almost 4.5% (for P@1 metric) in the Sharepoint dataset.

Since Stack Exchange reputation outperformed best answer ratio on the Programmers dataset, we were interested in the distribution of calculated reputation among the community. We provide a comparison between the best variant of our method (i.e. the full variant that calculates reputation for answering as well for asking questions, and eliminates an influence of amount of users' activity) and Stack Exchange reputation rules. In order to eliminate a long tail problem with reputation distribution in Programmers CQA system, we decided to group reputation by range of two (0-1, 2-3, etc.) and cut the high end of reputation.

The charts in Fig. 1 and Fig. 2, which contain histograms of reputation distributions calculated by our method and Stack Exchange reputation system respectively, clearly show that we were able to distinguish between the expertise of users better. Reputation calculated by our method follows Gaussian distribution what is expected, since we can naturally presume the majority of users to have average skills and knowledge. Another advantage of our approach is that we are able to better identify users with negative reputation.

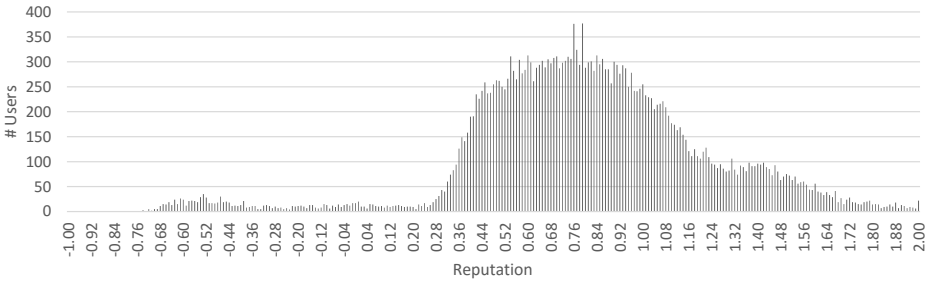

Fig. 1. Distribution of reputation calculated by our method

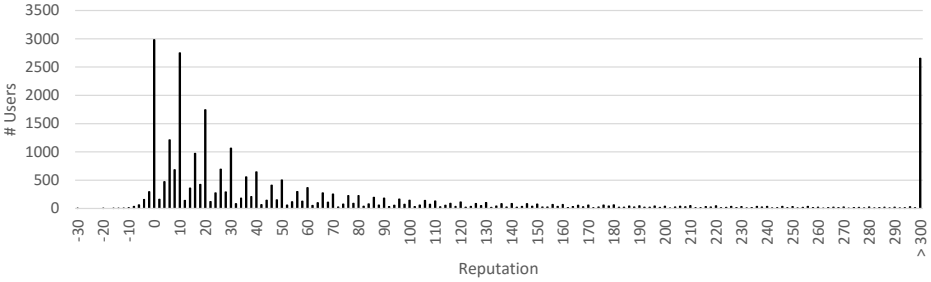

Fig. 2. Distribution of reputation calculated by Stack Exchange reputation system

Overall computational complexity of our method is the same or similar as for the previous reputation metrics or reputation schemas. However, as our method normalizes values of users contributions, it does not provide so good transparency for the end users as simple rule-based reputation schemas (e.g. they cannot easily verify why and how much of their reputation changed because they do not have simple access to all information required to make the calculation). Finding an optimal balance between precision and transparency of methods for user reputation calculation provides an interesting direction for further research.

5 Conclusions

In this paper, we introduced a method for estimating user reputation in CQA systems. Our main goal was to strengthen the importance of quality of user's contributions when calculating reputation. It is done by employing question difficulty and utility of questions and answers. The performance of our method was compared with other feature-based approaches on two datasets gathered from CQA systems provided by Stack Exchange platform. Our method outperformed all baselines, and thus we can confirm our assumption that consideration of content quality and difficulty plays an important role in estimation of user reputation. Moreover, we evaluated the distribution of calculated reputation among the community. We found out that reputation calculated by our method follows a continuous spectrum of values and a naturally occurring distribution, what is in contrast with the distribution of reputation calculated by the standard Stack Exchange reputation schema.

Encouraged by our results, we applied our method for reputation estimation in the educational and organizational CQA system Askalot [8], where it is running in production environment since May 2015. After consideration of educational nature of the system and the need to preserve factor of user activity, we decided to use the variant of our method which utilizes the sum of reputations for all questions and answers a user contributed.

For future work, it would be possible to investigate the importance of question difficulty and question/answer utility on the performance of our method. We can do this by assigning weight parameters to each component and observe differences in the performance when adjusting these values. Another possibility to improve our method lies in using clustering algorithms to find topics in CQA systems and do not rely on tags a user provided. We could also utilize an advanced method for content quality evaluation instead of the votes from the community. The problem of missing reputation gained for questions with no answers (due to unavailable estimation of question difficulty) could be solved by using average values of time to solve in the question's topic.

Acknowledgment. This work was partially supported by grants. No. VG 1/0646/15, VG 1/0774/16 and KEGA 009STU-4/2014 and it is the partial result of collaboration within the SCOPES JRP/IP, No. 160480/2015.

References

1. Bosu, A., Corley, C.S., Heaton, D., Chatterji, D., Carver, J.C., Kraft, N.A.: Building reputation in stackoverflow: An empirical investigation. In: Proceedings of the 10th Working Conference on Mining Software Repositories. pp. 89–92. MSR '13, IEEE Press, Piscataway, NJ, USA (2013)
2. Hanrahan, B.V., Convertino, G., Nelson, L.: Modeling problem difficulty and expertise in stackoverflow. In: Proc. of the ACM 2012 Conference on Computer Supported Cooperative Work Companion. pp. 91–94. CSCW '12, ACM, New York, NY, USA (2012)
3. Jurczyk, P., Agichtein, E.: Discovering authorities in question answer communities by using link analysis. In: Proc. of the Sixteenth ACM Conference on Conference on Information and Knowledge Management. pp. 919–922. CIKM '07, ACM, New York, NY, USA (2007)
4. Liu, D.R., Chen, Y.H., Kao, W.C., Wang, H.W.: Integrating expert profile, reputation and link analysis for expert finding in question-answering websites. Inf. Process. Manage. 49(1), 312–329 (Jan 2013)
5. Liu, J., Song, Y.I., Lin, C.Y.: Competition-based user expertise score estimation. In: Proc. of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 425–434. SIGIR '11, ACM, New York, NY, USA (2011)
6. Movshovitz-Attias, D., Movshovitz-Attias, Y., Steenkiste, P., Faloutsos, C.: Analysis of the reputation system and user contributions on a question answering website: Stackoverflow. In: Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. pp. 886–893. ASONAM '13, ACM, New York, NY, USA (2013)
7. Paul, S.A., Hong, L., Chi, E.H.: Who is authoritative? understanding reputation mechanisms in quora. CoRR abs/1204.3724 (2012)
8. Srba, I., Bielikova, M.: Askalot: Community question answering as a means for knowledge sharing in an educational organization. In: Proc. of the 18th ACM Conf. Comp. on Computer Supported Cooperative Work. pp. 179–182. CSCW'15, ACM, New York, NY, USA (2015)
9. Srba, I., Bielikova, M.: Why stack overflow fails? preservation of sustainability in community question answering. Submitted, IEEE Software (2015)
10. Yang, J., Tao, K., Bozzon, A., Houben, G.J.: Sparrows and owls: Characterisation of expert behaviour in stackoverflow. In: User Modeling, Adaptation, and Personalization, Lecture Notes in Computer Science, vol. 8538, pp. 266–277. Springer International Publishing (2014)
11. Zhang, J., Ackerman, M.S., Adamic, L.: Expertise networks in online communities: Structure and algorithms. In: Proc. of the 16th International Conference on World Wide Web. pp. 221–230. WWW '07, ACM, New York, NY, USA (2007)