Paulina Adamska
Marta Juźwin

STUDY OF THE TEMPORAL-STATISTICS-BASED REPUTATION MODELS FOR Q&A SYSTEMS

Abstract *Q&A systems are becoming a vital source of knowledge in many different domains. In some cases, they are also associated with services which provide employers with important information regarding the expertise of its potential employees. Therefore, the reputation earned in such communities can be associated with better job opportunities, and its significance is increasing. However, in a community where there is no direct financial motivation for participation, a reputation score is not solely an expertise metric. It is also a powerful motivator for remaining an active community member. Regardless of this complexity, algorithms for calculating reputation scores need to be as easy to understand (and implement) as possible. Therefore, the designers of the Q&A reputation system often implement a set of fixed rules, to some extent trading quality for quantity. Our goal is to study whether (and how) temporal statistics of a Q&A website can be incorporated into its reputation system. We want the proposed mechanism to dynamically adjust the impact of a single-answer evaluation on the reputation of its producer. We would like the proposed model to accurately reflect the expertise of content producers.*

Keywords Q&A systems, reputation systems, expertise, online communities

Citation Computer Science 16 (3) 2015: 253–263

1. Introduction

Reputation systems are well-known and widely-applied mechanisms used to enforce certain rules in online communities. They are particularly important and studied in the context of e-commerce [1, 5, 7, 10]; however, various implementations of such mechanisms are also introduced to different types of services, such as those devoted to content sharing (like reddit[1]) or those supporting sharing knowledge (like Q&A services). In such communities, the role of reputation systems is twofold. First of all – they provide motivation for remaining an active provider of high-quality content. On the other hand, reputation also substitutes for an expertise metric in various types of ranks, and is therefore strongly associated with the credibility of a community member. Researchers have already tried to speculate which properties of the produced content may be good expertise indicators [4] and tried to propose alternative reputation algorithms. The accuracy of the proposed approaches (in terms of reflecting user expertise) was evaluated against some reference metric and compared to solutions implemented in real communities. Nevertheless, in such solutions, a single modification of a thread associated with a certain question usually requires multiple reputation score recalculations. To the best of our knowledge, there is no extensive study of models utilizing temporal statistics to dynamically self- adjust the impact of a single-answer evaluation on the reputation of its producer in the context of Q&A platforms. In this work, we study whether replacing the set of fixed rules with such self-adjusting values can potentially make a reputation score a better expertise metric. The proposed models do not require any reputation gain recalculations when statistics change, so they are still computationally simple and easy to understand. We compare the performance of the proposed approaches to the standard reputation system (implemented by Q&A websites from the Stack Exchange[2] family) as well as the collaboration-based approach proposed by McNally et al. [6].

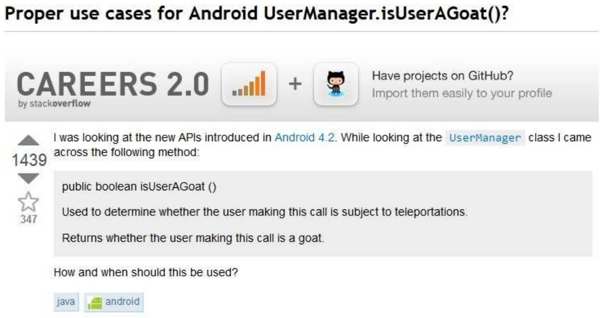
2. Related work

Researchers have already noticed that, in Q&A systems, the popularity of certain topics is the vital factor that influences the reputation of its contributors. This phenomenon refers both to the popularity of the general topic (for instance, a particular technology) and the popularity of the problem within the topic. The first aspect was mentioned by Bosu et al. [2], who discovered that expertise in some topics provides more opportunities to increase reputation by solving problems. However, unlike in our work, Bosu et al. measured popularity in terms of the number of questions. On the other hand, for each general topic, more- and less-common problems can be extracted. The first group yields more profits for contributors, but might not necessarily be correlated with the difficulty of a particular question, as complex issues often tend

[1] `http://reddit.com`

[2] `http://stackexchange.com/`

to be very specific or useful exclusively to experts. To mention some examples, there is a question on Stack Overflow (see Figure 1 that has received a high number of votes. This is obviously not posted to solve any important issue and does not require extensive knowledge nevertheless, it allows the respondent to significantly boost his reputation, even if the answer is not the first nor only one.



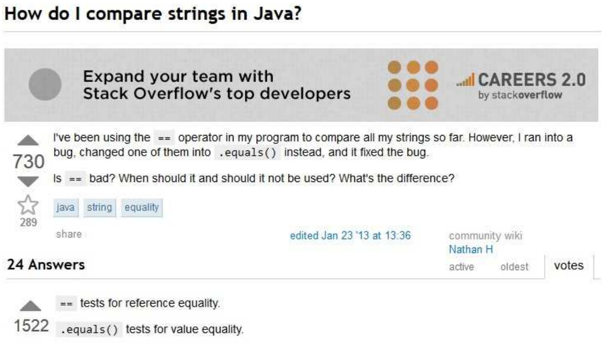
Proper use cases for Android UserManager.isUserAGoat()?

I was looking at the new APIs introduced in Android 4.2. While looking at the `UserManager` class I came across the following method:

> public boolean isUserAGoat ()
>
> Used to determine whether the user making this call is subject to teleportations.
>
> Returns whether the user making this call is a goat.

How and when should this be used?

Figure 1. Stack Overflow – first example of an extremely popular question.



How do I compare strings in Java?

I've been using the `==` operator in my program to compare all my strings so far. However, I ran into a bug, changed one of them into `.equals()` instead, and it fixed the bug.

Is `==` bad? When should it and should it not be used? What's the difference?

`==` tests for reference equality.

`.equals()` tests for value equality.

Figure 2. Stack Overflow – second example of an extremely popular question.

Apart from threads that have obviously been created for entertainment, there are also extremely popular posts that are associated with sharing very basic knowledge of a particular technology and can be answered with minimal effort, like the one regarding Java (presented in Figure 2).

The phenomenon of using upvotes to underscore the usefulness or entertainment value of posts makes it possible for community members to boost their reputations with minimal effort by actively posting easy yet very common questions, or by solving several common problems. The opportunity to utilize this mechanism is, to some extent, limited by the Community Wiki[3] feature, which disables automatic reputation

[3] http://meta.stackoverflow.com/questions/11740/what-are-community-wiki-posts

increments when certain content gets upvoted. However, in many cases, the decision whether a particular post should be owned by the entire community is made directly by its members, so it depends both on user honesty and the individual interpretation of the term.

There have been many proposals of a different set of rules for computing the reputation of users and estimating the quality of posts on Stackoverflow. Romano and Pinzger [9] propose an algorithm of calculating answer scores, which aims to address the problem of the first answer having an advantage over those that follow, due to the fact that the computed score does not depend on evaluation time, and some of the voters might not have seen all of the answers at the moment of making their evaluations. The proposed algorithm assigns different weights to the votes, depending on the number of answers already posted when the evaluation was made. However, this approach still allows users to exploit popular threads, and it has only been studied in regards to whether this approach makes any difference in the system (namely, whether such an algorithm would choose different best answers). No analysis of how it is associated with the quality of the chosen content has been made.

McNally et al. [6] studied the impact of a collaboration-based reputation model on Stack Exchange websites. For each answer with a score greater than zero, its producer received a trust score proportional to the number of upvotes (in comparison to the sum of scores of all answers in the particular thread). This reputation algorithm has been evaluated by analyzing its correlation with the "ground-truth reputation score", which was computed using the accepted answer rate for each answerer. This mechanism has also been proven to perform better than the original Stack Exchange approach (the original reputation score was normalized by the highest score in the system) as well as the Page Rank approach. The proposed solution helps to reduce the impact of more-popular questions and promotes the less-popular (and probably more costly to answer) ones, as they yield a better opportunity to maximize the potential gain of having even only one upvote. However, this method has one disadvantage – it requires the reputation of each question answerer to be recomputed every time a vote is submitted to any of the answers in the particular thread. This yields potential problems, both for the system designer (most of the existing web services choose to avoid such complexities and prefer to predefine a fixed set of possible reputation modification options, which can be manually tuned if it is necessary) and user experience (which is crucial for communities that rely on members volunteering to submit posts). For many users, it may be important to know that their reputation can only be decreased when his or her content gets downvoted and that it cannot be affected by changes of other answerer evaluations.

3. Experimental reputation systems

In this paper, we evaluate two experimental models. The first one (also referred to as **temporal question score-based**) adjusts the weight of each answer upvote using the temporal score of the associated question at the time of vote arrival. The reputation increment for answerer u, after receiving an upvote for the answer to question q at time t, is computed according to the following formula:

$$\text{reputationgain}_{(u)} = \frac{1}{|\text{score}_{(q,t)}| + 1} \tag{1}$$

where $|\text{score}_{(q,t)}|$ is the absolute value of the score of question q_y (for which the answer is provided) at time t (which is the time of receiving an answer upvote). We use absolute value, assuming that a good answer can also be provided for a downvoted question. This model assumes that questions with a higher score are very often common issues that are easy to solve. Therefore, they do not necessarily indicate high expertise and should not be associated with an extremely high reputation gain.

The second model (also referred to as temporal answer number-based) is very similar; however, it utilizes a temporal answer number at the time of vote arrival. This approach should be more resilient to adversaries, as it takes more effort to manipulate the number of answers than to control the question score using the voting mechanism.

The reputation increment for answerer u, after receiving an upvote for the answer to question q at time t, is computed according to the following formula:

$$\text{reputationgain}_{(u)} = \frac{1}{|\text{answers}_{(q,t)}| + 1} \tag{2}$$

where $|\text{answers}_{(q,t)}|$ is the number of answers to question q (for which the evaluated answer is provided) at time t (which is the time of receiving an answer upvote). This model assumes that multiple answers for a single question may indicate that more users have some knowledge of the topic. Thus, such issue may be easier to solve and should not be associated with an extremely high reputation score.

The proposed mechanisms do not introduce any punishment for submitting a down-voted answer. Both of them also assume that we do not update reputation gains when temporal statistics (used to compute them) change.

4. Analysis

In this section, we describe the dataset used in our analysis along with the research methodology. In the last subsection, we present the preliminary results.

4.1. Dataset

For the purpose of our analysis, we have extracted the publicly-available data regarding the activity of users over one year from StackOverflow, which is one of the most popular Q&A for programmers. The basic statistics regarding the dataset are presented in Table 1. Event history used to compute user reputation according to different algorithms consisted of post submissions (both questions and answers) and the chosen post evaluation activities (namely upvotes, downvotes, and answer acceptances).

Table 1

Analyzed dataset statistics.

Portal	Number of posts	Number of votes	Time span	Number of users
Stackoverflow	922750	2704272	1 year	37171

4.2. Methodology

In order to provide some quality metric of the proposed approaches, we needed to choose some reference metric. We have chosen the collaboration-based approach. The main reason for our choice was that this approach does not promote popular topics. There are a fixed number of reputation points to win (namely, one point in the original proposal) for providing an answer to each question. The reputation gain of each answerer depends on the share of upvotes compared to the total number of upvotes in a particular thread. Such an environment makes it more profitable to answer more-complex questions (as they may get fewer answers) and therefore maximize the chance of winning 100% of the points offered for answering a single question. Moreover, it does not depend on additional answer acceptances and allows better solutions to eventually be promoted regardless of the time of posting an answer.

We have used the previously-described event trace and computed the reference reputation score using the original collaboration-based algorithm. Both reference and experimental reputation scores were normalized using the maximum reputation generated by the studied mechanism to keep all of the computed values within the $\langle 0, 1\rangle$ range. After that, reference values were sorted in descending value and matched with the corresponding values computed using the evaluated algorithms. Thus, we were able to compare the final score of each user for both the reference and experimental algorithms. Following McNally et. al. [6], we have computed Spearman correlations between the normalized reference reputation scores in our experiments, and the experimental ones for the best n users from the reference rank, were $n \in$ 50, 100, 150, . . . , 500, all.

To additionally estimate the performance of the studied approaches, we have also computed a benchmark reputation score, which was the standard StackOverflow approach. In our calculations of the StackOverflow reputation scores, we have only considered answer upvotes (increasing reputation by 10 points), answer downvotes (decreasing reputation by 2 points) and answer acceptances (increasing reputation by 15 points). We did not implement any additional rules (like reputation decrease for downvoters, as there is no information about the voter in the publicly-available dataset).

4.3. Reference metric considerations

In order to provide some quality metric of the collaboration-based approach, we have used the previously-described event trace and computed reference reputation scores using two different algorithms. The first one is a simple mechanism based on the answer acceptance rate, as proposed by McNally et al. [6]; however, we made one important modification. This approach assigns higher weight to answers to those questions that have received less attention from the community, and therefore received fewer answers. Assuming A_{q_i} being the entire set of answers to question i at the end of the observed time period, the reputation score in this case is increased by $\frac{1}{|A_{q_i}|}$ for each accepted answer. This metric allows us to capture answers that have indeed solved the problem stated by the question asker, and it is independent of topic popularity, as there is only one point of reputation to gain for providing an answer to a single question. Moreover, it rewards solving problems which may be potentially more specialized, more difficult, or require additional effort (ex. some experiments). Such intuition is supported by Pal et al. [8], who have verified the hypothesis that experts tend to answer lower-value questions where higher value yields a higher total answer score in a particular thread and the presence of an accepted answer.

To some extent, the first reference metric incorporates question difficulty. For each accepted answer, its author receives a number of points proportional to the total number of answers to a particular question at the end of the observed time period. This intuition is supported by the assumption that, the more users are able to provide any solution to the problem (regardless its quality) or believe to have some knowledge on the topic, the more common an issue is considered in this particular thread. On the other hand, Hanrahan et.al. [3] proposed using the duration between the time of posting a question and answer acceptance to estimate question difficulty. The quality of this difficulty metric has not been verified in any way by the authors of this proposal, and may have two potential drawbacks. It is unclear to what extent the time needed to post an accepted answer estimates the actual difficulty of the solved issue, and to what extent it is the estimator of some personal traits of the question asker. We can imagine a community member who is extremely fastidious and always waits for the most-accurate answer, another person who accepts even a partial solution if it gives some good ideas quickly enough, or perhaps even a third person who rarely uses the accept option at all. Nevertheless, we have decided to take such a reference metric into consideration as well. In our analysis, we were not able to extract the exact acceptance time from the public datadump (only the date was provided). Instead, we chose to use the time needed to provide an answer that was eventually accepted. If there was no accepted answer to a particular question available in the dataset, we calculated the time between question posting and the date of the last post available in the database. Later on, the times were normalized and used to compute the potential new reference scores. For each answer acceptance, a community member could increase his reputation by the amount of points proportional to the time between posting the question and submitting the accepted answer.

To additionally estimate the performance of the collaboration-based approach, we have also computed a benchmark reputation score (which was the standard StackOverflow approach). Both reference and experimental reputation scores were preprocessed as described in section 4.2.

The results for the first reference metric show that the performance of the collaboration-based approach is significantly better than the Stack Overflow approach, not only for the simple acceptance-rate-based approach (as described in the original paper) but also for our weighted acceptance rate (see Figure 3). The results for the second reference metric are presented in Figure 4. Interestingly, we can observe that, for the group of best users, the correlations for both metrics are comparatively similar and extremely poor. Nevertheless, the collaboration-based reputation performance seems to be as good as the Stack Overflow approach.

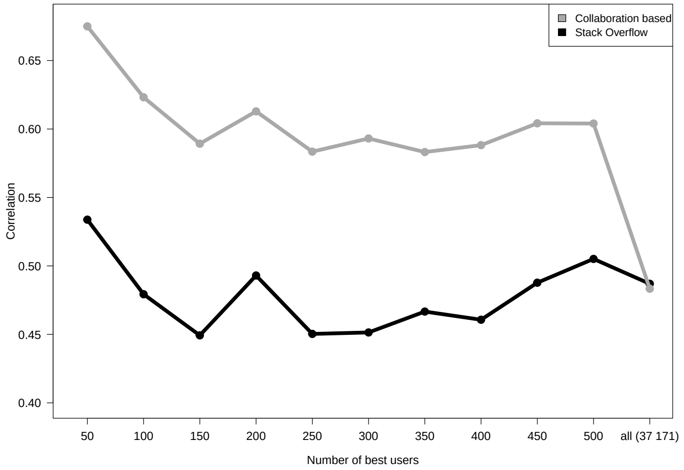


Figure 3. Weighted acceptance rate used as the reference reputation.

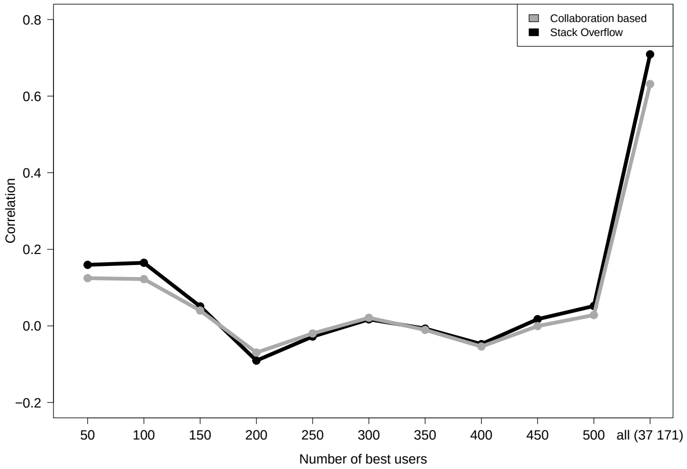


Figure 4. Acceptance time used as the reference reputation.

For evaluation purposes, we assume, that the behavior of honest content evaluators is not altered by the new rules, as the proposed reputation score only affects the reputations of question answerers. Adversary-resilience analysis is outside the scope of this paper.

4.4. Experimental approaches evaluation

In our experiments, we have compared the benchmark approach and our experimental reputation models to the collaboration-based reference score. The results for both experimental reputation systems are depicted in Figure 5. As we can see, the temporal question score-based approach is more successful in mimicking the collaboration-based rank than the original StackOverflow approach. We can observe even better results in the case of the second experimental mechanism (based on temporal answer number). This improvement can be associated with one of the features of the reference metric; namely, its indirect dependency on the answer number (the fewer different answers available, the better chance of getting the largest reputation gain due to a lack of competition). As previously mentioned, the temporal answer number-based model is also potentially better in terms of adversary resilience, as it is more costly to manipulate this statistic.

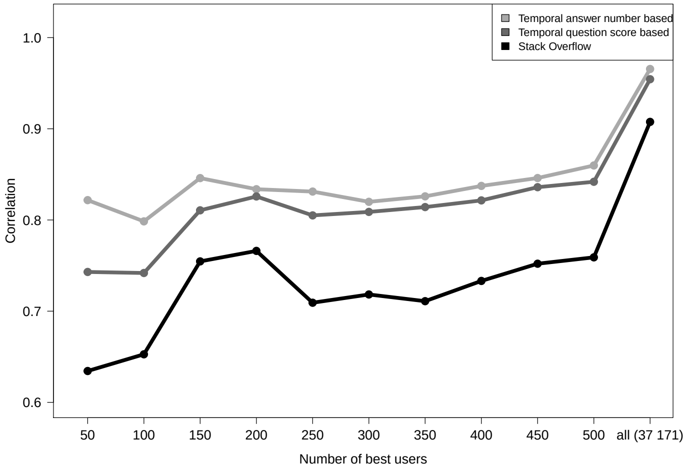


Figure 5. Performance of the proposed reputation models.

5. Conclusions and future work

According to our preliminary results, it seems to be possible to incorporate temporal statistics into Q&A reputation systems. We have shown that such an approach, although extremely simple, can also be surprisingly accurate with respect to our reference score. We are planning to further verify this hypothesis using data from different

Q&A websites and investigate some more sophisticated reputation models involving temporal statistics, along with their resilience to adversaries. We would also like to verify the effectiveness of various reputation mechanisms in terms of user-expertise approximation, utilize different reference scores (and take into account manually-evaluated difficulty levels of a chosen set of questions), and check the computational complexity of the proposed models.

Acknowledgements

This work is supported by Polish National Science Center grant 2012/05/B/ST6/03364.

References

[1] Borzymek P., Sydow M., Wierzbicki A.: Enriching Trust Prediction Model in Social Network with User Rating Similarity. In: Ajith Abraham, Vaclav Snasel, eds, *Proceedings of the 1st International Conference on Computational Aspects of Social Networks (CASoN 2009)*, pp. 40–47, IEEE Computer Society, Los Alamitos, NY, USA, 2009.

[2] 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*, MSR '13, pp. 89–92. IEEE Press, Piscataway, NJ, USA, 2013, http://dl.acm.org/citation.cfm?id=2487085.2487107.

[3] Hanrahan B. V., Convertino G., Nelson L.: Modeling problem difficulty and expertise in stackoverflow. In: *CSCW '12 Computer Supported Cooperative Work, Seattle, WA, USA, February 11–15, 2012 – Companion Volume*, pp. 91–94, 2012, http://dx.doi.org/10.1145/2141512.2141550.

[4] Kao W., Liu D., Wang S.: Expert finding in question-answering websites: a novel hybrid approach. In: *Proceedings of the 2010 ACM Symposium on Applied Computing (SAC), Sierre, Switzerland, March 22-26, 2010*, pp. 867–871, 2010, http://dx.doi.org/10.1145/1774088.1774266.

[5] Kaszuba T., Hupa A., Wierzbicki A.: Advanced Feedback Management for Internet Auction Reputation Systems. *IEEE Internet Computing*, vol. 14(5), pp. 31–37, 2010, http://dx.doi.org/10.1109/MIC.2010.85.

[6] McNally K., O'Mahony M. P., Smyth B.: A Model of Collaboration-based Reputation for the Social Web. In: *Proceedings of the Seventh International Conference on Weblogs and Social Media, ICWSM 2013, Cambridge, Massachusetts, USA, July 8–11, 2013*, 2013, http://www.aaai.org/ocs/index.php/ICWSM/ICWSM13/paper/view/6112.

[7] Morzy M., Wierzbicki A.: The Sound of Silence: Mining Implicit Feedbacks to Compute Reputation. In: *Internet and Network Economics, Second International Workshop, WINE 2006, Patras, Greece, December 15–17, 2006, Proceedings*, pp. 365–376, 2006, http://dx.doi.org/10.1007/11944874_33.

[8] Pal A., Harper F. M., Konstan J. A.: Exploring Question Selection Bias to Identify Experts and Potential Experts in Community Question Answering. *ACM Transation Information Systems*, vol. 30(2), p. 10, 2012, http://dx.doi.org/10.1145/2180868.2180872.

[9] Romano D., Pinzger M.: Towards a Weighted Voting System for Q&A Sites. In: *2013 IEEE International Conference on Software Maintenance, Eindhoven, The Netherlands, September 22–28, 2013*, pp. 368–371, 2013, http://dx.doi.org/10.1109/ICSM.2013.49.

[10] Wierzbicki A.: The Case for Fairness of Trust Management. *Electron. Notes Theor. Comput. Sci.*, vol. 197(2), pp. 73–89, 2008, http://dx.doi.org/10.1016/j.entcs.2007.12.018.

Affiliations

Paulina Adamska
Polish-Japanese Academy of Information Technology, Warsaw, Poland, tiia@pjwstk.edu.pl

Marta Juźwin
Polish-Japanese Academy of Information Technology, Warsaw, Poland,
marta.juzwin@pjwstk.edu.pl

Received: 19.01.2015
Revised: 20.03.2015
Accepted: 20.03.2015