

SwiftQ: Nearly Real-time Voluntary Question-Answering Platform

Chae Ryn Chang
Daejeon, Korea
KAIST
chaec0803@kaist.ac.kr

Joon Young John Chung
Daejeon, Korea
KAIST
johnr0@kaist.ac.kr

Kyung Je Jo
Daejeon, Korea
KAIST
kyungjejo@kaist.ac.kr

ABSTRACT

Online platforms such as StackOverflow, and Quora successfully generate knowledge through crowdsourcing and help users in their information needs. However, these platforms naturally encounter challenges in delayed response time, lack of participation, and quality of responses. We present SwiftQ, a nearly real-time voluntary question-answering platform. Our platform addresses challenges many existing platforms naturally encounter through voluntary crowdsourcing. We conducted a user study to verify the usability and effectiveness of our prototype. We find the organic environment where the user is both a requester and a respondent can both ensure high quality of responses and nearly real-time responses.

Author Keywords

Question-answering, online community, voluntary crowdsourcing, mobile application

INTRODUCTION

The high accessibility to the internet worldwide has successfully led to new forms of knowledge production on a scale never seen before. In addition, due to its convenience, reliance on various online websites to acquire new knowledge is now becoming heavier for all users from employees at work to families at home to students at a school. In consequence, the use of online sites such as Wikipedia, StackOverflow, or Quora is more pervasive now than ever.

Different online platforms produce and collect knowledge in varying ways. For example, Wikipedia collects knowledge in a less dynamic method organized on topic-basis. Multiple workers collaborate to generate a page full of information on a specific topic. Question and answer (Q&A) community (e.g., StackOverflow and Quora) takes more dynamic method allowing users to upload questions of their own interest, and then other users voluntarily answer the questions. With either approach, online platforms produce knowledge at scale that is shareable to a global population online.

While large number of users benefit from knowledge in various online platforms, the drawbacks are often overlooked. 1) Due to a large number of users, the quality of knowledge generated by others is not always well controlled. 2) Many questions are not answered real-time or remain unanswered forever. 3) Platforms require users to answer questions voluntarily which lead to a poor participation; otherwise, the platform is not for free. 4) Existing platforms are not so interactive providing one-way responses, often leading to requester's poor understanding.

We present SwiftQ¹, a nearly real-time voluntary question and answer platform. Users are provided nearly real-time responses for free from other users who are the experts in the domain of the uploaded question. It is an organic platform where the users are both a requester and a respondent simultaneously. We conducted a user study to evaluate the usability and effectiveness of our system. The result suggests that quality responses and supporting explanations can be generated using the platform.

RELATED WORKS

For a long time, there has been research on improving the Q&A site's reliability. One challenge is that validity of answers cannot always be ensured. Jurczyk et al. and Zhang et al. addressed the problems by identifying the level of expertise of the answers [6, 10]. Other works relied on community-based ratings in order to ensure the quality of answers [1, 2, 3]. In this paper, instead of analyzing the answers, we address the problems by only allowing qualified users to enter the platform.

Previous works on Q&A community analyzed user behavior thoroughly. Welser et al. [7] proved that there exists unbalanced share of answering by some users. Other work by Zhang et al. [10] examined an online platform to show that there are users that ask, users that answer, and users that do both. Adamic et al. [1] analyzed Yahoo! Answers and found huge separation between the requester and the respondent. Not only having qualified answers but also having any answers in the first place is a critical part in QA platforms. Narrowing the gap between requester and respondent is remaining as a challenge.

Prior research has uncovered many of the motivations and incentives for answering questions in online community. Yu et al. [9] discussed motivations such as active learning, self-enhancement, reciprocity, reputation, enjoyment of helping others, self-protection, moral obligation and the advancement of the virtual community. Contant et al. [4] examined how answers are produced without no explicit reward. The work found altruism is a strong motivation for answering questions. While works by Harper et al. [5] and Yang et al. [8] suggested that offering monetary reward can induce high quality of answers, it showed little effect in participation rate.

¹<https://www.youtube.com/watch?v=b140bleVrfw>

DESIGN GOALS

From problems of existing question-answering platforms, we identified the following three design goals.

1. Assure reliable answers from high quality workers

In all question-answering platforms, achieving high quality of answer is the most important factor: with low assurance of the quality, people would not use the question-answering platform. Thus, to keep the question-answering platform to be robust and to retain users within the platform, the efficient method of quality control is necessary.

2. Allow users to get answer quickly

In existing question-answering systems that make the use of crowds' capability to answers questions, requester often needs to experience a long delay before they receive answers. Moreover, in platforms such as StackOverflow, many problems remain unanswered. Such factor demotivates people from using the question-answering system. To allow users to maximize the time efficiency in online interaction, the platform must ensure a time-saving method to keep response time as quick as possible.

3. Make users motivated to answer questions

In the voluntary crowdsourcing system, giving users a reason to participate is crucial. Within our system, motivating workers is also a key part because they need to answer questions quickly. Even though previous works on real-time crowdsourcing have worked on retaining workers with the paid method to facilitate the real-time crowdsourcing, non-monetary reward approach has been researched less. Therefore, ways to motivate workers without money and instead using other forms of rewards are necessary to make the nearly real-time question answering system.

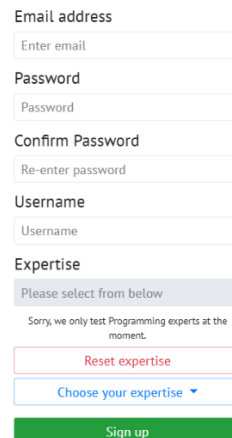
SYSTEM

SwiftlyQ is a nearly real-time crowdsourcing platform on which crowds contribute to the platform by exchanging their expertise by answering questions in their expertise and getting answers to questions which are not of their expertise. Initially, the system decides the expertise of workers by testing them with gold standard problems. Within the system, each user will have at least an expertise. SwiftlyQ continuously notifies users whenever a new question in user's expertise is uploaded by other users. Furthermore, the system motivates crowds to dedicate themselves to it by offering achievements if their performance meets conditions for them and by giving mediums to exchange their expertise, tokens.

Pre-Screening Workers with Gold Standard Question

In this question-answering system, making qualified people work on questions of their expertise is important to assure the quality of the answer. For that, the system decides whether people are qualified to answer questions when they sign up to the system. When users sign up to the system, they can choose their own expertise from options offered, and in the prototype, they can choose Physics, Calculus or Programming. Before finishing the sign-up, they are offered gold standard questions that are adequate to their selection of expertise, which are intended to test their expertise. If they do pass the test of the

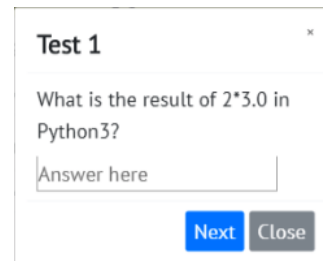
SwiftlyQ



The sign up page form includes the following fields and buttons:

- Email address: Enter email
- Password: Password
- Confirm Password: Re-enter password
- Username: Username
- Expertise: Please select from below. A message says: "Sorry, we only test Programming experts at the moment." Below this are buttons for "Reset expertise" and "Choose your expertise" (a dropdown menu).
- Sign up: A large green button at the bottom.

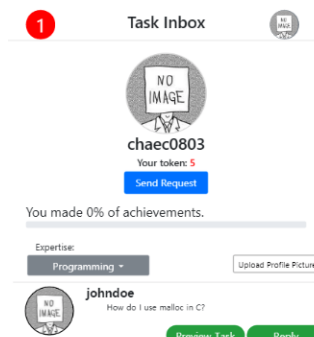
Figure 1: Sign up page



The question interface shows:

- Test 1
- What is the result of $2^*3.0$ in Python3?
- Answer here
- Next and Close buttons

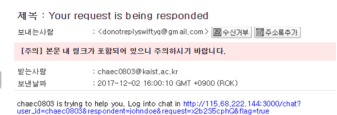
Figure 2: Gold standard question for programming expertise



The Task Inbox page shows a user profile for 'chaec0803' with 5 tokens and a 'Send Request' button. Below the profile, it says 'You made 0% of achievements.' and 'Expertise: Programming'. There is an 'Upload Profile Picture' button. At the bottom, there are 'Preview Task' and 'Reply' buttons.

Figure 3: Inbox page

Figure 4: Request notification



The request notification email content includes:

- 제목 : Your request is being responded
- 보내는사람 : <donotreply@swiftlyq.com> [공식신호] [주소복합기]
- [주의] 본문 내 링크가 포함되어 있으니 주의하시기 바랍니다.
- 발신시간 : chaec0803@fast.ac.kr
- 발신날짜 : 2017-12-02 16:00:10 GMT +0900 (KROK)
- chaec0803 is trying to help you. Log into chat in http://115.68.222.144:3000/chat?user_id=chaec0803&respondnotofloodrequest=cb295c9fca87a947ae

Figure 5: Answer notification

domain, they are qualified to participate as workers and can solve problems of the domain for others. If they do not pass, they are not authorized to solve problems of the domain.

Nearly Real-Time Question-Answering Session

The question-answering session begins with posting of a question by a requester. If a user has a question which she wants to know how to solve, she can post the question with the domain of the question specified. She can describe the problem not only with text explanation but also by uploading related pictures. After the requester submits the request on the system, a notification e-mail is sent to users who have related expertise. For instance, if a requester uploads a programming problem, all the users with programming expertise will receive the notification. Experts then can go to the inbox to initiate a chat with the person who posted the question. Then a notification is sent to the requester, and if the requester reacts to the notification, the answering session begins.

In the answering session, the question and the related picture are shown to both of the requester and the expert. They

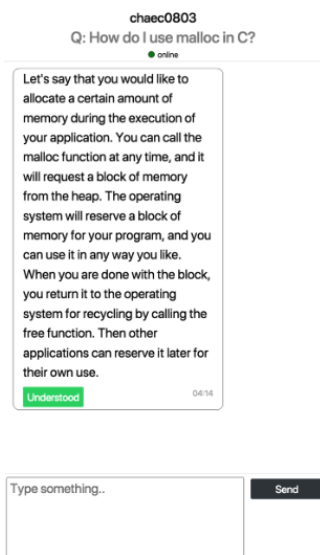


Figure 6: Question-answering session interface

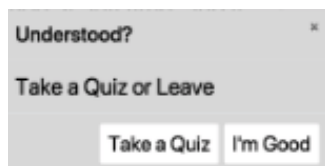


Figure 7: Pop up page that appears when the requester hits the 'understood' button

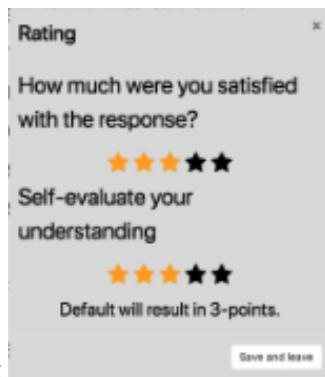


Figure 8: Rating interface shown to the requester

can basically communicate with the chat interface. They can communicate until the requester understands how to solve the problem, and if the requester finally understands, she can press 'understood' button to signify that she now can solve the problem alone. The 'understood' button is intended to make the answering session be centered on the requester and persist until the requester is satisfied.

After the requester understood the problem, if she wants to test herself on the topic, she can request a quiz to the expert. If so, the expert has to give the requester quiz through the chat interface. When the requester gets correct on the quiz, the expert can signify that the requester is correct. After that, the requester should leave a rating to the expert in 5-star scale, which is used for the achievement system and the reputation of workers. Then she can leave the question-answering page. The expert can leave the page after she signifies that the requester got correct on the quiz, and if she meets a condition for an achievement, she will receive the achievement. If the requester does not want the quiz, the requester can leave the rating rightly and leave the question-answering page, and the expert can also leave the page rightly.

Achievement and Token System

To motivate participants, the system offers them achievements if they meet conditions for them. Once they satisfy the condition, the achievement pop-up is shown to them. One goal of achievements is to promote agile react of users to new request uploaded in the system. For instance, they can get an achievement if they can solve a problem in 5 minutes after the request is uploaded, which requires user's swiftness to the new request. Also, the system intends to make workers

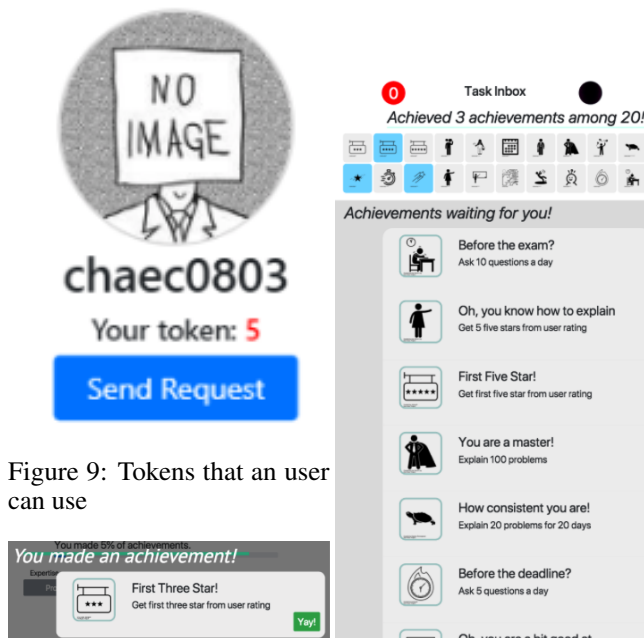


Figure 9: Tokens that a user can use

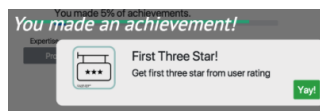


Figure 10: Achievement pop up shown to the expert when met a condition of an achievement

Figure 11: Achievement panel

be more reliable with achievements. For example, if workers receive many five stars from requesters, they are rewarded with an achievement. Users can view what they achieved so far from the achievement view panel, which promotes workers to achieve more achievements.

We also implemented the token system to assure that no user is only taking advantage of the system, only requesting and getting answers. All users are basically receiving five tokens in the beginning and each token can be used for requesting a question. When users explain how to solve problems for others, they receive tokens as the reward, and the amount of reward is decided by which rating they got: they receive more tokens if they get higher the rating.

USER STUDY

We conducted a user study in the laboratory setting, where we divided the role of experts and requesters. We recruited nine requesters and three workers, and let them freely conduct question-answering sessions for one day, only requiring requesters to post at least one question and workers to answer three questions. As a result of the study, fifteen question-answering sessions were conducted. All questions were answered successfully.

We collected reaction of users by conducting informal interviews with them. In overall, requesters seem to be satisfied with explanations they got from experts. One of the requesters told us that he could learn a lot from the expert, complimenting the mathematical capability of the expert and his kindness

in explaining topics that the requester could not understand well. Experts also showed positive reactions to the system, especially for features that motivate people who explain. One of the users pointed out the achievement panel and said that she really liked the achievement system and panel because they have many features, which would keep experts away from getting bored with the task.

However, participants also pointed out limitations of the system. First of all, they pointed out the lack of robust quality control methods. In the current implementation of the system, the only coercive quality control measurement is gold standard questions given when signing up. Even though there is a rating system which reflects the reputation of users, it was not extensively used in that it was not visible to other users. Because of these reasons, users showed worries about the possibility of being trolled when they ask questions.

Also, the limitation was found in the notification system. Because of the limitation of the framework we used, the notification to users could only be sent by e-mails, which is not a direct notification to users. Because of this limitation, the real-time feature could not be fully supported, delaying users when they try to be connected with requesters or experts. Also, there were minor irritations with the implementation, like people complaining about notification e-mail piling in their e-mail inboxes.

CONCLUSION & DISCUSSION

Through user study, SwiftyQ suggests that voluntary crowdsourcing can lead to nearly-real time responses in relatively assured quality. Though we tested the basic functionality of the system for question-answering, the user study was limited in that we could not study the extensive effects of the system. The test of the system was only for a day. The dynamics of users in the system can differ if their usage continues for a longer time period. On top of that, because we only collected a small number of participants, we could not verify whether the system really supports real-time question-answering in the wild condition.

Even though we found that nearly real-time question-answering is effective in explaining problems in a personalized way, sometimes the session did not begin rightly. In this case, the stored question-answer data might be more efficient. Because the nearly real-time question-answering session and the stored question-answer set both have their own benefits and shortcomings, building a system that balances benefits of both might be a possible future work.

REFERENCES

1. Lada A. Adamic, Jun Zhang, Eytan Bakshy, and Mark S. Ackerman. 2008. Knowledge Sharing and Yahoo Answers: Everyone Knows Something. In *Proceedings of the 17th International Conference on World Wide Web (WWW '08)*. ACM, New York, NY, USA, 665–674. DOI: <http://dx.doi.org/10.1145/1367497.1367587>
2. Eugene Agichtein, Carlos Castillo, Debora Donato, Aristides Gionis, and Gilad Mishne. 2008. Finding High-quality Content in Social Media. In *Proceedings of the 2008 International Conference on Web Search and Data Mining (WSDM '08)*. ACM, New York, NY, USA, 183–194. DOI: <http://dx.doi.org/10.1145/1341531.1341557>
3. Jiang Bian, Yandong Liu, Eugene Agichtein, and Hongyuan Zha. 2008. Finding the Right Facts in the Crowd: Factoid Question Answering over Social Media. In *Proceedings of the 17th International Conference on World Wide Web (WWW '08)*. ACM, New York, NY, USA, 467–476. DOI: <http://dx.doi.org/10.1145/1367497.1367561>
4. David Constant, Lee Sproull, and Sara Kiesler. 1996. The Kindness of Strangers: The Usefulness of Electronic Weak Ties for Technical Advice. *Organization Science* 7, 2 (1996), 119–135.
5. F. Maxwell Harper, Daphne Raban, Sheizaf Rafaeli, and Joseph A. Konstan. 2008. Predictors of Answer Quality in Online Q&A Sites. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. ACM, New York, NY, USA, 865–874. DOI: <http://dx.doi.org/10.1145/1357054.1357191>
6. Pawel Jurczyk and Eugene Agichtein. 2007. Discovering Authorities in Question Answer Communities by Using Link Analysis. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management (CIKM '07)*. ACM, New York, NY, USA, 919–922. DOI: <http://dx.doi.org/10.1145/1321440.1321575>
7. Howard T. Welser, Eric Gleave, Danyel Fisher, and Marc Smith. 2007. Visualizing the Signatures of Social Roles in Online Discussion Groups. *The Journal of Social Structure* 8, 2 (2007). <http://www.cmu.edu/joss/content/articles/volume8/Welser/>
8. Jiang Yang, Lada A. Adamic, and Mark S. Ackerman. 2008. Crowdsourcing and Knowledge Sharing: Strategic User Behavior on Taskcn. In *Proceedings of the 9th ACM Conference on Electronic Commerce (EC '08)*. ACM, New York, NY, USA, 246–255. DOI: <http://dx.doi.org/10.1145/1386790.1386829>
9. Jie Yu, Zhenhui Jiang, and Hock Chuan Chan. 2007. Knowledge Contribution in Problem Solving Virtual Communities: The Mediating Role of Individual Motivations. In *Proceedings of the 2007 ACM SIGMIS CPR Conference on Computer Personnel Research: The Global Information Technology Workforce (SIGMIS CPR '07)*. ACM, New York, NY, USA, 144–152. DOI: <http://dx.doi.org/10.1145/1235000.1235034>
10. Jun Zhang, Mark S. Ackerman, and Lada Adamic. 2007. Expertise Networks in Online Communities: Structure and Algorithms. In *Proceedings of the 16th International Conference on World Wide Web (WWW '07)*. ACM, New York, NY, USA, 221–230. DOI: <http://dx.doi.org/10.1145/1242572.1242603>