Noname manuscript No.
(will be inserted by the editor)

Bounties on Technical Q&A Sites: A Case Study of Stack Overflow Bounties

Jiayuan Zhou · Shaowei Wang · Cor-Paul Bezemer · Ahmed E. Hassan

Received: date / Accepted: date

Abstract Technical question and answer (Q&A) websites provide a platform for developers to communicate with each other by asking and answering questions. Stack Overflow is the most prominent of such websites. With the rapidly increasing number of questions on Stack Overflow, it is becoming difficult to get an answer to all questions and as a result, millions of questions on Stack Overflow remain unsolved. In an attempt to improve the visibility of unsolved questions, Stack Overflow introduced a bounty system to motivate users to solve such questions. In this bounty system, users can offer reputation points in an effort to encourage users to answer their question.

In this paper, we study 129,202 bounty questions that were proposed by 61,824 bounty backers. We observe that bounty questions have a higher solving-likelihood than non-bounty questions. This is particularly true for long-standing unsolved questions. For example, questions that were unsolved for 100 days for which a bounty is proposed are more likely to be solved (55%) than those without bounties (1.7%).

In addition, we studied the factors that are important for the solving-likelihood and solving-time of a bounty question. We found that: (1) Questions are likely to attract more traffic after receiving a bounty than non-bounty questions. (2) Bounties work particularly well in very large communities with a relatively low question solving-likelihood. (3) High-valued bounties are associated with a higher solving-likelihood, but we did not observe a likelihood for expedited solutions.

Jiayuan Zhou · Shaowei Wang (✉) · Ahmed E. Hassan
Software Analysis and Intelligence Lab (SAIL), Queen's University, Kingston, ON, Canada
E-mail: jzhou, shaowei, ahmed@cs.queensu.ca

Cor-Paul Bezemer
Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada
E-mail: bezemer@ualberta.ca

Our study shows that while bounties are not a silver bullet for getting a question solved, they are associated with a higher solving-likelihood of a question in most cases. As questions that are still unsolved after two days hardly receive any traffic, we recommend that Stack Overflow users propose a bounty as soon as possible after those two days for the bounty to have the highest impact.

1 Introduction

Online technical Q&A sites have become more and more important for software developers to share knowledge. Developers can post questions on such Q&A sites and receive answers from other developers. Stack Overflow[1] is a prominent example of such a Q&A site. Stack Overflow has more than 16.8 million questions, 25.9 million answers, and 9.7 million users.[2]

Stack Overflow has become an important source on which developers rely to solve various software engineering problems (Ahasanuzzaman *et al.*, 2018; Wu *et al.*, 2018). For example, developers post questions on Stack Overflow about their programming problems, in the hope of receiving helpful responses. However, with the rapid increase of the number of questions on Stack Overflow, solving all the questions has become a challenge for the community. Although many of the questions on Stack Overflow are solved quickly (the median waiting time is less than one hour (Wang *et al.*, 2018c)), 47.2% (8,023,388) of the questions are not solved at all.[3]

Some customer-driven crowd-sourcing marketplaces, such as Fenda (China)[4] and Whale (US)[5], introduced monetary incentives to motivate users to make contributions (Jan *et al.*, 2017). In contrast, Stack Overflow uses gamification in the form of a point-based reputation system to motivate users to make a contribution (e.g., answering questions or revising posts). To motivate users through gamification, Stack Overflow introduced a bounty system. Through this bounty system, *bounty backers* can offer reputation points by proposing a bounty for the user who answers a question. Although bounties have been used since January 2009,[6] the association between bounties and the solving-likelihood and solving-time of a question have never been examined. By understanding this association, we could provide insights on how to better leverage bounties to solve questions.

In this paper, we perform a large-scale analysis of the bounty system of Stack Overflow by studying 129,202 bounty questions that were proposed by 61,824 bounty backers. We first conduct a preliminary study in which we

[1] https://stackoverflow.com/
[2] https://data.stackexchange.com/
[3] https://data.stackexchange.com/stackoverflow/query/968466
[4] https://fd.zaih.com/fenda
[5] https://techcrunch.com/2016/10/31/justin-kan-launches-video-qa-app-whale/
[6] https://stackoverflow.blog/2009/01/27/reputation-bounty-for-unanswered-questions/

uncover that bounty questions have a higher solving-likelihood than non-bounty questions. We show that bounties work particularly well for solving long-standing unsolved questions, and for solving questions in very large communities with a relatively low question solving-likelihood.

In addition, we study in depth which factors are important for the solving-likelihood and solving-time of bounty questions. Finally, we study the impact of bounties on the traffic to questions. The main findings of our study are as follows:

1. Questions are likely to attract more traffic after receiving a bounty than non-bounty questions, particularly when the value of the bounty is high (i.e., 400). In addition, bounty questions have a higher solving-likelihood than non-bounty questions, especially for questions in very large communities with a relatively low question solving-likelihood.
2. Bounty questions with a higher bounty value have a higher solving-likelihood, however, a higher bounty value does not expedite a bounty question getting solved.
3. Bounty questions tend to have a higher solving-likelihood than non-bounty questions, particularly when focusing on long-standing unsolved questions. For example, questions that were unsolved for 100 days for which a bounty is proposed are more likely to be solved (55%) than those without a bounty (1.7%).

Our study shows that while bounties are not a silver bullet for getting a question solved, they are associated with a higher solving-likelihood in most cases. As questions on Stack Overflow generally are not solved at all if they remain unsolved after two days, we recommend that users post their bounty as soon as possible after these two days.

Paper Organization. The rest of this paper is organized as follows: Section 2 presents background information about Stack Overflow and its bounty system. Section 3 describes our data collection process. Section 4 describes our preliminary study. Sections 5 and 6 describe our model for studying the factors that are associated with the solving-likelihood and the solving-time of bounty questions. Section 7 studies the relation between bounties and the traffic of bounty questions. Section 8 studies special cases of bounties and discusses the implications of our study. Section 9 discusses threats to validity of our study. Section 10 introduces related work. Finally, Section 11 concludes our study.

2 Background

2.1 The Question and Answer Process on Stack Overflow

Stack Overflow is one of the largest software developer communities in the world, with more than 50 million software developers using it every month. Developers ask and answer questions on Stack Overflow, and they can upvote or downvote answers and questions to reflect their opinions. The score of a post

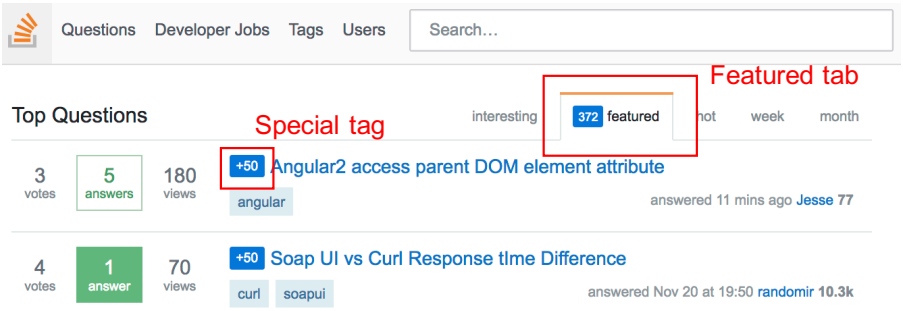


Fig. 1: A screenshot of Stack Overflow's "featured" tab which highlights bounty questions.

is the sum of its up and down votes. Each question may have many answers, but only one answer can be accepted by the asker as the accepted answer. When a question gets an accepted answer, the question is *solved*.

Stack Overflow uses several gamification features, such as the reputation system, to motivate the members of its community to interact with each other through these questions and answers. For example, a user gains reputation points if the user's posts (i.e., questions or answers) receive upvotes from others. We can approximate how the question asking and answering-skills of a developer are perceived by the Stack Overflow community by looking at their reputation points. There are good reasons for users to have a good reputation on Stack Overflow. For example, Stack Overflow profiles are sometimes used during the recruitment process by software companies (such as Stack Overflow itself) as a measure of the technical knowledge of a developer. In addition, Stack Overflow users get elevated privileges, such as a reduced number of displayed ads on the website, as their reputation grows (Stack Overflow, 2019).

There exist two ways to consume reputation points: (1) by proposing bounties for a question to attract more attention from the community or to reward an existing answer; (2) by downvoting a post. In this study, we focus on the first way, in which users consume reputation points by proposing bounties.

2.2 The Bounty System on Stack Overflow

Stack Overflow has a bounty system that allows users to offer reputation points for any user that would produce an accepted answer to a question, in an effort to draw more attention from users across the site. Figure 2 shows the life cycle of a bounty. When a user asks a question, anyone can propose a bounty on that question after two days, thereby becoming a *bounty backer*. A question can only have one active bounty at any time. In other words, one cannot propose another bounty on a question if the question already has an active bounty at that moment. Note that when a bounty is proposed, the reputation points that

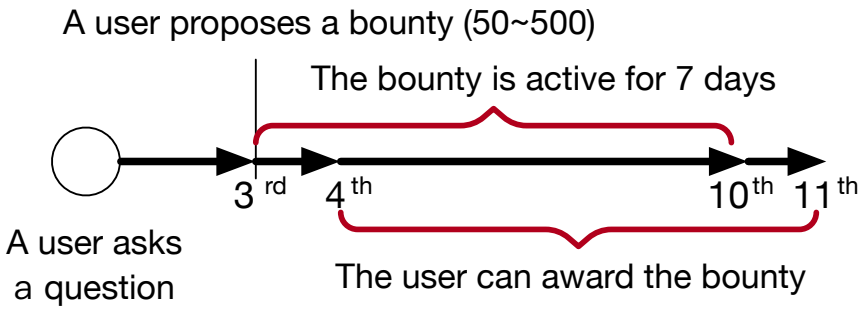


Fig. 2: The life cycle of a bounty.

are offered in the bounty are removed immediately from the bounty backer's reputation (and are never refunded even if the question remains unsolved at the expiry of the bounty).

Users can propose a bounty with a value between 50 and 500 reputation points, in 50-point increments. A bounty can be active for a maximum of seven days. While a bounty is active, the bounty question is labeled with a special tag that highlights its associated bounty value. The question itself is highlighted in the "featured" tab on the Stack Overflow homepage (see Figure 1) to help draw attention from the community towards that question.

A bounty can be awarded to an answer by the bounty backer one day after it was proposed. If the bounty backer does not explicitly award the bounty, it will be automatically awarded one day after the expiry date of the bounty. The rules for the automated awarding of bounties are:[7]

1. If the bounty backer is the original question asker, the bounty will be awarded to the answer that was accepted while the bounty was active.
2. An answer that was created after the bounty was offered which has more than one vote (but was not accepted) will be awarded half of the bounty value. If there are multiple answers that meet this criterion, the bounty is awarded to the earliest answer.
3. If no answer meets the above two criteria the offered reputation points are discarded.

When a bounty question gets an answer which is awarded the bounty, we define the bounty question as *solved*. Note that the awarded answer can also be an answer which is not accepted by the question asker.

3 Data Collection

StackExchange (Stack Exchange, 2017) provides a Stack Exchange Data Dump,[8] which is composed of a set of XML files that contain data about all questions,

[7] https://stackoverflow.com/help/bounty
[8] https://archive.org/details/stackexchange

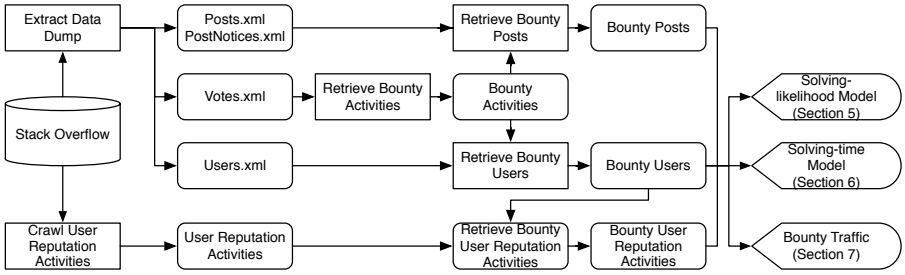


Fig. 3: An overview of our data collection process.

Table 1: Dataset description.

Period	Sep. 23, 2011 to Aug. 27, 2017
Number of bounty questions	129,202
Number of expired bounties	44,635
Number of bounty backers	61,824
Number of non-bounty questions	12,359,663

answers, tags, votes, and user histories of Stack Overflow. We use the following files from this set:

1. `Posts.xml` contains data about posted questions and answers.
2. `PostNotices.xml` contains the reasons for offering each bounty.
3. `Votes.xml` contains data about activities, such as the date on which a question was upvoted. `Votes.xml` also contains bounty activity information. For example, the creation and closure date of a bounty, the bounty value, the id of the related user who proposed or won the bounty, and the id of the related question or answer.
4. `Users.xml` contains data about users, such as the user id, the creation date of their accounts, and their reputation at the time of the data archival.

Figure 3 gives an overview of our data collection process. We first downloaded the data dump of Aug. 27, 2017. Because the last major change to Stack Overflow's bounty system was made on Sep. 23, 2011,[9] we only study bounties that were proposed between Sep. 23, 2011 and Aug. 27, 2017. Then we collected the data as follows:

1. We first retrieved the bounty activity information from `Votes.xml`. Then we retrieved the posts and users that are associated with the selected bounties accordingly.
2. We crawled the history of reputation activities from each user's profile page on Stack Overflow,[10] and we traced back their reputation activities to the moment of proposing a bounty.

[9] https://stackoverflow.blog/2011/09/23/bounty-reasons-and-post-notices/
[10] https://stackoverflow.com/users/userid?tab=reputation

We observed that for some bounty questions, the available data about the life cycle of the bounty is incomplete. For example, the question "How to detect which one of the defined font was used in a web page?"[11] only shows when the bounty was rewarded, but not when it was created. After removing the bounty questions with an incomplete bounty life cycle from our data, our dataset contains 129,202 bounty questions, which involved 61,824 bounty backers who proposed bounties, and 12,359,663 non-bounty questions. Table 1 gives an overview of our studied dataset.

There exist several 'special' cases of bounties, in which the bounty was used for a purpose other than getting a question solved. To avoid bias in our study, we treat the following cases separately:

1. Bounties that were proposed to reward existing answers. Such bounties can be filtered easily as the bounty was created with the reason "Reward existing answer".
2. Bounties that were automatically awarded by Stack Overflow. A bounty that was awarded automatically does not reflect that the bounty backer is satisfied with the answer.
3. Bounties that were proposed while the question already had an accepted answer. For example, a bounty was offered with the purpose of drawing attention to a question on April 14, 2012.[12] However, the bounty was eventually awarded to the answer that was already given on April 6, 2012.

We discuss the first case in more detail in Section 8. In the second case, we cannot distinguish whether the bounty question was actually solved, as the rewarded answer is not necessarily a solution to the question. The third case is difficult to recognize automatically, as we cannot distinguish between whether the bounty backer wanted to reward the existing answer, or was looking for additional answers. Hence, we remove bounties of these types and their associated questions from our dataset. Note that we keep all unsolved bounty questions for which the bounty expired. After separating the special bounty cases, our dataset contains 79,093 bounty questions.We published our data online.[13]

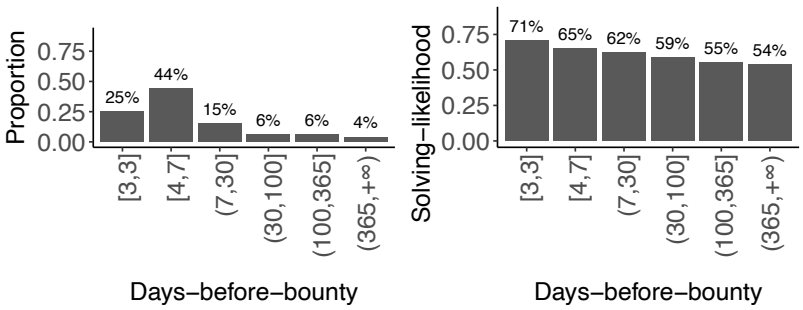
4 Preliminary Study

In our preliminary study, we first present basic descriptive statistics about bounties from the following perspectives: (1) the solving-likelihood of a question, (2) the number of days between the creation of a question and the proposal of its first bounty (i.e., the *days-before-bounty* metric), (3) the solving-time of a bounty question after the bounty is proposed, and (4) the bounty

[11] https://stackoverflow.com/questions/845/

[12] https://stackoverflow.com/posts/10038098/revisions

[13] https://github.com/SAILResearch/supportmaterial-18-jiayuan-SO_bounty/tree/master/data_model



(a) (b)

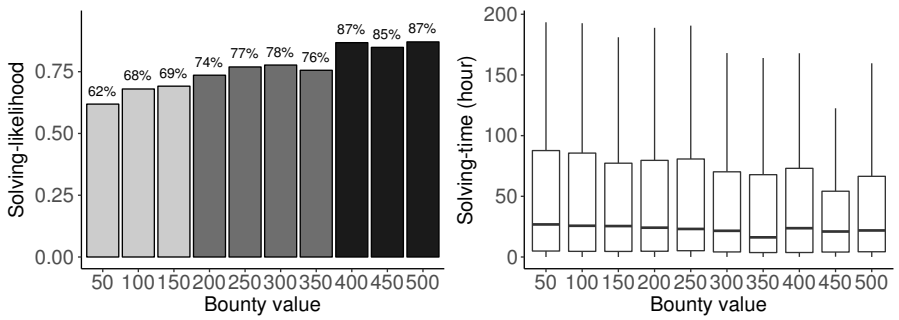
Fig. 4: (a) The proportion of bounty questions across different values of the days-before-bounty metric. (b) The solving-likelihood of bounty questions across different values of the days-before-bounty metric.

value. From these statistics, we get a basic overview of bounties on Stack Overflow. Second, we discuss the impact of bounties on the solving-likelihood of bounty questions across Stack Overflow tags.

In general, bounty questions have a higher solving-likelihood than non-bounty questions. The solving-likelihood of bounty questions is 65.5% which is 30% higher than that of non-bounty questions (i.e., 48.9%). Especially for the questions with more than one bounty, the solving-likelihood is 92.0%.

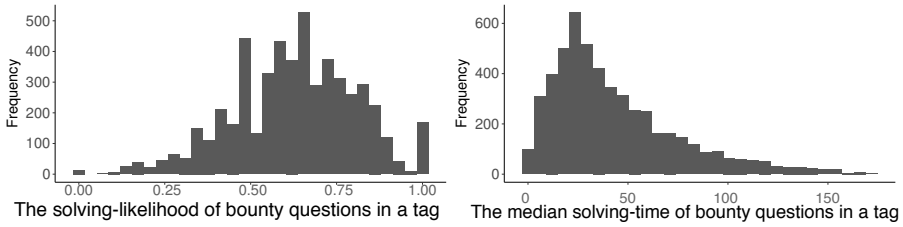
Long-standing unsolved questions with bounties are more likely to be solved than those without bounties. Prior work (Anderson *et al.*, 2012) showed that questions either get solved very quickly, or not at all. Figures 4a and 4b show the proportion and solving-likelihood of bounty questions for different values of the days-before-bounty metric. 69% of the bounties were proposed within one week while only 10% of the bounties were proposed after 100 days since the creation of a question. However, the solving-likelihood for such"late bounty questions" is around 55% (i.e., 2,605 out of 4,776 questions). In comparison, only 104,831 out of 6,321,124 (1.7%) of the non-bounty questions that were unsolved 100 days after their creation were solved afterwards. Hence, long-standing unsolved questions with bounties are more likely to be solved than those without bounties.. The time after which a bounty is proposed is related the solving-likelihood: 25% of the bounties were proposed three days after the creation of the question, which is the earliest date on which it is allowed to propose a bounty. The solving-likelihood of these bounties is the highest (i.e., 71%).

Bounty questions with higher-valued bounties have a higher solving-likelihood. However, higher bounty values are not associated with expedited solutions. Figure 5a shows the solving-likelihood of bounty questions across different bounty values. In general, the solving-likelihood increases as the bounty value increases. In particular, there is a large difference in



(a) Solving-likelihood (b) Solving-time

Fig. 5: The distribution of the solving-likelihood and the solving-time of bounty questions across different bounty values. The bars are marked with different shades to indicate the levels of solving-likelihood that we distinguished.



(a) Solving-likelihood (b) Solving-time

Fig. 6: The distribution of the solving-likelihood and solving-time of the tags of bounty questions. Each data point in the distribution represents one tag.

solving-likelihood (62.9% vs. 88.1%) between the lowest and highest bounty values (50 and 500).

We grouped the bounty values into three groups (showed by different shades in Figure 5a) that correspond to partitions of 10% of the solving-likelihood (i.e., 60% to 70%, 70% to 80% and 80% to 90%) for our study in Section 7.

Figure 5b shows the solving-time of bounty questions for different bounty values. Counter-intuitively, we do not observe a clear association between the bounty value and the solving-time. The correlation between the solving-time and value is -0.02 which indicates a weak association.

Bounty questions have a higher solving-likelihood than non-bounty questions. Bounties appear to work especially well for long-standing unsolved questions. Bounty questions with a higher bounty value have a higher solving-likelihood. However, there is no association between a bounty's value and its solving-time, which implies that a higher bounty value does not expedite the solving of a bounty question.

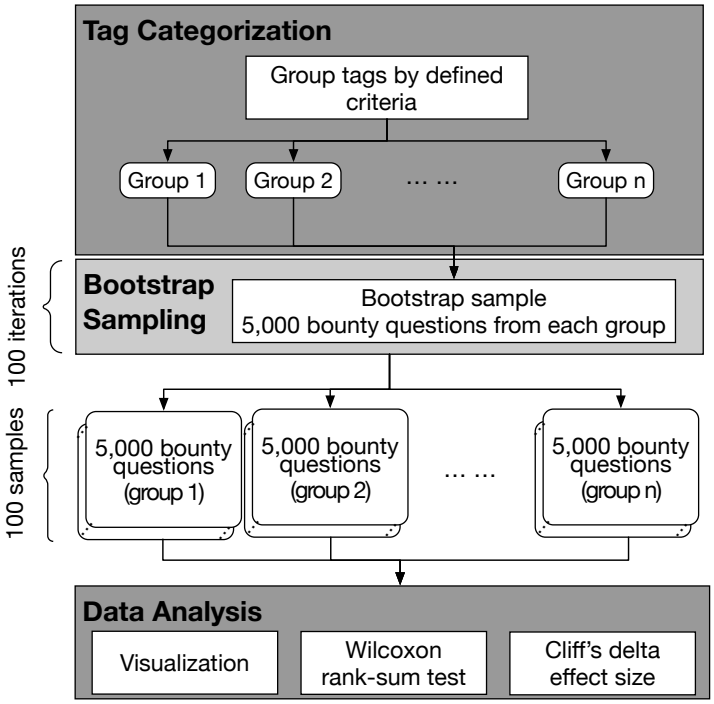


Fig. 7: Overview of our approach for studying the relation between bounties and the solving-likelihood of bounty questions across tags.

The solving-likelihood and the solving-time of a bounty question varies across tags. When posting a question, the question asker can assign one or more tags to the question to attract more targeted traffic. However, some of these tags are more popular than others. To reduce the bias caused by tags which have only a few bounty questions, we only consider the tags which have more than five bounty questions for the following two figures. Figures 6a and 6b show the frequency of tags in terms of the solving-likelihood and solving-time of bounty questions, respectively. We observed that the solving-likelihood and solving-time of bounties vary across tags. For example, the solving-likelihood of bounty questions with the `applescript-studio` tag is 70%, while the solving-likelihood of questions with the `xcode9-beta` tag is 40%. In the remainder of this section, we look in more detail into the impact of bounties on the solving-likelihood of bounty questions across tags.

4.1 The Association between Bounties and the Solving-likelihood of Bounty Questions across Tags

Figure 6a shows that the solving-likelihood of bounty questions differs across tags. In this section, we study how bounties impact the solving-likelihood of

Table 2: The distribution of 20,180 bounty-related tags across the size and skill-based groups.

Size-based Categorization			**Community-quality-based Categorization**		
Group Name	**#Tags**	**#Questions**	**Group Name**	**#Tags**	**#Questions**
Small	16,540	47,457	Micro	904	1,416
Moderate	3,182	78,519	Small	10,507	114,567
Large	439	94,320	Medium	8496	166,339
Very Large	19	62,607	High	273	582

bounty questions across answerer communities (i.e., tags) of different sizes and with varying question solving-likelihoods. The population of answerers within a community indicates the size of the community.

We first grouped all tags by their size (*size-based*) and question solving-likelihood (*community-quality-based*). Then we used a bootstrap sampling approach to sample tags and questions in each group in order to ensure the statistical stability of our observations. Finally, we studied the solving-likelihood of questions across the size-based and community-quality-based groups. Figure 7 gives an overview of our approach. We detail each step below.

Step 1: Tag categorization. Since the answerer population for tags ranges from 1 to 386,885, we grouped the tags into four size-based groups according to the order of magnitude of their answerer population. We created the community-quality-based groups by grouping the tags according to their solving-likelihood for non-bounty questions in intervals of 0.25. To summarize, the tags were grouped based on the following criteria:

Criteria for size-based categorization:

- **Small:** The answerer population of a tag is smaller than 1,000.
- **Moderate:** The answerer population of a tag is between 1,000 and 10,000.
- **Large:** The answerer population of a tag is between 10,000 and 100,000.
- **Very large:** The answerer population of a tag is larger than 100,000.

Criteria for community-quality-based categorization:

- **Micro:** The tag's non-bounty question solving-likelihood is less than 0.25.
- **Small:** The tag's non-bounty question solving-likelihood is between 0.25 and 0.50.
- **Medium:** The tag's non-bounty question solving-likelihood is between 0.50 and 0.75.
- **High:** The tag's non-bounty question solving-likelihood is more than 0.75.

Table 2 shows the distribution of tags across the size and skill-based groups. To reduce the bias that is caused by the unbalanced number of tags and questions across groups, we employed bootstrap sampling.

Step 2: Bootstrap sampling. We applied a bootstrap sampling approach to sample bounty questions of each size and skill-based group. We first randomly sampled 5000 tags from each group with replacement. Then we randomly sampled one bounty question from each sampled tag, to reduce the bias towards

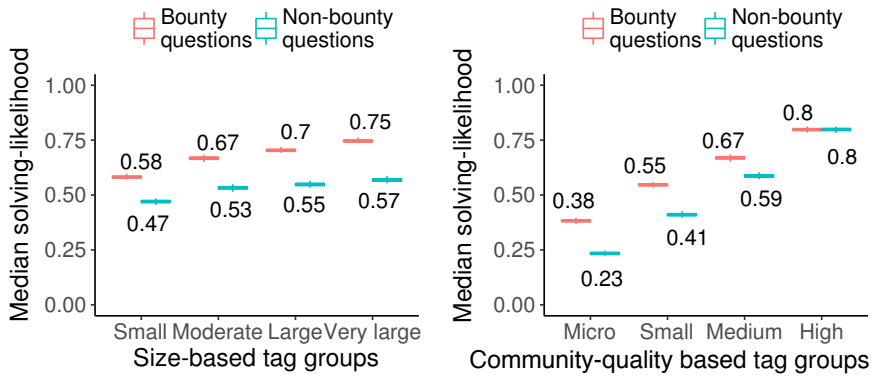


Fig. 8: The distribution of the median solving-likelihood across the size-based and skill-based tag groups for the 100 studied samples for bounty and non-bounty questions.

tags with more bounty questions. Hence, we sampled 5,000 bounty questions for each group. To ensure the statistical robustness of our results, we repeated our bootstrap sampling process 100 times with different random seeds. We ended up with 100 samples and for each sample, there are 20,000 bounty questions for the size-based groups and 20,000 for the skill-based groups (5,000 bounty questions for each group).

Step 3: Data analysis. For each sample, we calculated the solving-likelihood across the size and skill-based groups. To compare the differences between two distributions, we used the Wilcoxon rank-sum test (Bauer, 1972), which does not require the distribution to be normally distributed. We also performed the Bonferroni correction (Bonferroni, 1936) to correct the p-values for multiple comparisons. Furthermore, we applied Cliff's delta d effect size (Long *et al.*, 2003) to quantify the magnitude of the differences. We use the following thresholds for d (Romano *et al.*, 2006): $|d| \leq 0.147$ (negligible); $0.147 < |d| \leq 0.33$ (small); $0.33 < |d| \leq 0.474$ (medium); $0.474 < |d| \leq 1$ (large).

Results: **The solving-likelihood of bounty questions is higher than that of non-bounty questions across all size-based tag groups.** Figure 8 shows the distribution of the solving-likelihood of bounty and non-bounty questions across the size-based tag groups. For all size-based groups, the solving-likelihood of bounty questions is significantly higher than that of non-bounty questions (with a large effect size). The solving-likelihood of both bounty and non-bounty questions increases as the size of the tag group gets larger. A possible explanation is that a large community has more answerers, so there is a higher chance of someone solving the bounty.

The solving-likelihood of bounty questions is higher than that of non-bounty questions that are asked in communities with a lower question solving-likelihood. Figure 8 shows the distribution of the solving-likelihood of bounty and non-bounty questions across the community-quality-

based tag groups. The solving-likelihood of bounty questions is higher than that of non-bounty questions that are asked in different community-quality-based groups except the 'High' group. A possible explanation is that as the solving-likelihood in the 'High' group is already very high (80%), it is hard to improve – the unsolved questions may be too hard or unclear to answer. We also observe a few tag outliers in which the solving-likelihood of bounty questions is lower than that of non-bounty questions while still having many bounties. For example, the "*flash-builder*" tag has 50 bounty questions although the solving-likelihood of its bounty questions is 0.26, which is much lower than its non-bounty questions (i.e., 0.53). One possible reason is that the bounty questions under this tag are very specific and require specific knowledge, which not many people possess.

> *The solving-likelihood of bounty questions is higher than that of non-bounty questions, especially in very large communities with relatively low question solving-likelihood.*

In the next sections, we build logistic regression models to further study the important factors that are associated with the solving-likelihood and solving-time of a bounty question.

5 What Are the Important Factors that Are Associated with the Solving-likelihood of a Bounty Question?

In our preliminary study, we observed an association between the solving-likelihood of a bounty question and two bounty-related factors (i.e., the bounty value and the days-before-bounty metric). We also noticed that the solving-likelihood of bounties differs across tags. There may be other factors that impact the solving-likelihood of a bounty question. For example, longer bounty questions with code snippets may have a higher solving-likelihood. In this section, we use a model to study other factors that may have a relation with the solving-likelihood of bounty questions. With a better understanding of this relationship, we can provide insights into how to better leverage bounties to improve the solving-likelihood of questions.

5.1 Approach

We built a logistic regression model to study the relationship between the studied factors and the solving-likelihood of bounty questions. The logistic regression is a robust and highly interpretable technique, which has been applied successfully in several software engineering-related problems (Wang *et al.*, 2018c; McIntosh *et al.*, 2016). The primary goal of our model is not to determine whether a bounty question will be solved, but to better understand the relationship between each factor and the likelihood of a bounty question

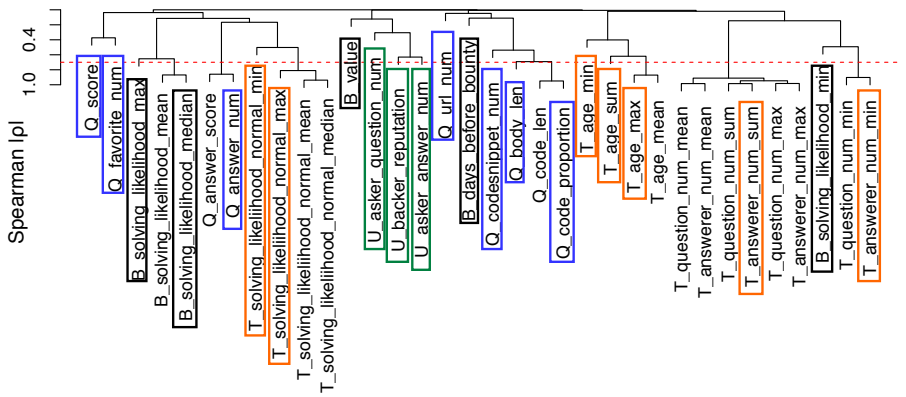


Fig. 9: The hierarchical clustering plot of factors in our solving-likelihood model. According to the Spearman rank correlation test (using a cut-off value of 0.7), we selected the simplest metrics to compute across each dimension of correlated factors. We ended up with seven factors in the question level dimension (marked in blue), three factors in the user dimension (marked in green), five factors in the bounty dimension (marked in black), and seven in the tags dimension (marked in orange).

being solved. We are the first to study the relation between the studied factors and the solving-likelihood of a bounty question, hence we expect future work to expand on our studied factors. In the following subsections, we elaborate on the studied factors, the details of the model construction, and the analysis of our model.

5.1.1 Studied Factors

We study factors along the following dimensions:

1. **Question:** Nine factors which reflect the quality of a question and the activities that are related to the question.
2. **User:** Three factors which reflect the reputation of the bounty backer and the question asker.
3. **Bounty:** Six factors which describe the usage (e.g., the value) of bounties of the question and its associated tags.
4. **Tag:** 16 factors which reflect the community of a bounty question in terms of the age, the answerer population and the question-solving skills of the answerer (i.e., the non-bounty question solving-likelihood).

We have 34 factors in total. Table 3 shows the description of and rationale for the 34 studied factors. These factors are explanatory variables of our model.

Table 3: The description of and rationale for the factors that we used in our logistic regression model for the solving-likelihood of bounty questions. The factors which are marked with '*' are calculated at the time when the bounty is proposed and the factors which are marked with '**' are calculated considering only the data one month before the bounty is proposed.

Factor name	Description	Rationale
The Question Dimension		
Q_url_num	The number of URL links in the content of a question.	These factors reflect the amount of supportive information that a question has. Questions with more supportive information may help potential answerers in solving.
Q_codesnippet_num	The total number of code snippets in the content of a question.	
Q_body_len	The length of the content of a question (in characters).	
Q_code_len	The total length of the code snippets the content of a question (in characters).	
Q_code_proportion	The proportion of code the content of a question (i.e., $\frac{Q\_code\_len}{Q\_body\_len}$).	
Q_answer_score *	The total score of all answers of a question.	These factors reflect the popularity of a question and its answers. Popular questions may attract more attention.
Q_answer_num *	The number of answers that a question received.	
Q_score *	The score of a question.	
Q_favorite_num *	The favorite count of a question.	
The User Dimension		
U_backer_reputation *	The reputation of the backer who proposed the bounty.	A backer with a high reputation may indicate that the backer is a knowledgeable user and the question may be of high quality.
U_asker_answer_num **	The number of prior answers of the question asker.	A question, which is asked by an asker whose prior activity is high, may be of high quality.
U_asker_question_num **	The number of prior questions of the question asker.	
The Bounty Dimension		
B_days_before_bounty	The number of days between the creation of a question and the proposing of a bounty for it.	The timing of proposing a bounty may have a relationship with the solving-likelihood.
B_value	The value of a question's bounty.	A higher bounty may attract more potential answerers.
B_solving_likelihood_-min/max/mean/median *	The min/max/mean/median solving-likelihood of bounty questions for a question's tags.	
The Tag Dimension		
T_age_min/max/mean/sum	The min/max/mean/sum age in days of a question's tags.	Older tags may have a larger community and more potential answerers to solve questions
T_question_num_-min/max/mean/sum **	The min/max/mean/sum number of questions of a question's tags.	These factors reflect the community size of the tags of a question. A larger community size may have more potential answerer to solve questions.
T_answerer_num_-min/max/mean/sum **	The min/max/mean/sum number of answers of a question's tags.	
T_solving_likelihood_normal_-min/max/mean/median *	The min/max/mean/sum age of a question's tags.	These factors reflect the question-solving skill level of answerers of a bounty question's associated communities. Questions that have communities with highly skilled answerers are more likely to be solved.

5.1.2 Model Construction

The presence of correlated and redundant features (i.e., multicollinearity) negatively impact the interpretability of the generated classifiers (Farrar and Glauber, 1967; Tantithamthavorn and Hassan, 2018). Therefore, similar to prior studies (Rajbahadur *et al.*, 2017; Wang *et al.*, 2018c; McIntosh *et al.*, 2016), we first removed correlated and redundant factors to avoid multicollinearity. We used the Spearman rank correlation test to measure the correlation between the studied factors and kept only one of the highly-correlated factors (using a cut-off value of 0.7 (McIntosh *et al.*, 2016; Thongtanunam *et al.*, 2016; Wang *et al.*, 2018c; Tantithamthavorn and Hassan, 2018)). Then we conducted a redundancy analysis to remove redundant factors using `R`'s `redun` function. Finally, we ended up with seven factors in the question dimension, three factors in the user dimension, five factors in the bounty dimension, and seven factors in the tags dimension (Figure 9). We built a logistic regression model, which enables us to examine the impact of one or more variables on a response variable while controlling for other variables. Similar to prior work (McIntosh *et al.*, 2016; Wang *et al.*, 2018c), we added non-linear terms in the model to capture the more complex relationship in the data by employing restricted cubic splines (Harrell, 2006). The non-linear factor will be assigned additional degrees of freedom (i.e., `D.F.`). We used the **rms**[14] `R` package to implement our logistic regression model. See our prior work (Wang *et al.*, 2018c) for more details about the non-linear term allocation.

5.1.3 Model Analysis

We used the Area Under the ROC Curve (i.e., AUC) and a bootstrap-derived approach (Efron, 1986) to assess the explanatory power of the logistic regression model following prior studies (McIntosh *et al.*, 2016; Wang *et al.*, 2018c). The AUC ranges from 0 to 1 (with 0.5 being the performance of a random guessing model). A higher AUC indicates that the model has a better ability to capture the relationship between the explanatory variables and the response variable. In the bootstrap-derived approach, we built a model with a bootstrapped sample then we applied the model to the original dataset and the bootstrapped dataset. We used the optimism value, which is the difference of the AUC between the models that are built on the original data and the model that is built on the bootstrapped dataset, to evaluate the amount of overfitting. Small optimism values indicate that the model does not suffer from overfitting. We repeated the bootstrap-derived approach 100 times and used the median optimism value to evaluate the overfitting of our models.

To understand the impact of each factor in the model, we used the `anova` function in the `R` package `rms` to compute the Wald χ^2 value (i.e., the importance of a factor) and the statistical significance (p-value) of each factor. We also apply a Bonferroni correction (Bonferroni, 1936) to correct the p-values

[14] https://cran.r-project.org/web/packages/rms/index.html

for multiple comparisons. We used the `Predict` function in the `rms R` package to plot the estimated bounty question solving-likelihood against a factor. The analysis allows us to further carefully examine how each factor affects the solving-likelihood. We hold the other factors at their median values when exploring one factor.

Table 4: The result of our logistic regression model for understanding the relationship between the studied factors and the bounty question solving-likelihood. The factors are ordered by their importance (i.e., overall Wald's χ^2 value) in the model. We also show the non-linear (NL) Wald χ^2 value. We only show factors of significant importance (i.e., the p-value of the χ^2 value is less than 0.002 (i.e., 0.05/22)) to our model. See our supplementary material for the full table (Zhou, 2019).

Factors		**Solving-likelihood Model**	
AUC		0.708	
AUC optimism		0.001	
Factors		**Overall**	**NL**
Q_answer_num	D.F.	1	
	χ^2	1348.604	
B_value	D.F.	9	
	χ^2	597.668	
T_solving_likelihood_normal_min	D.F.	4	3
	χ^2	473.843	7.921
B_days_before_bounty	D.F.	1	
	χ^2	382.611	
T_answerer_num_sum	D.F.	2	3
	χ^2	359.326	108.199
T_solving_likelihood_normal_max	D.F.	3	2
	χ^2	349.808	54.763
B_solved_likelihood_median	D.F.	4	3
	χ^2	164.312	128.110
B_solved_likelihood_min	D.F.	3	2
	χ^2	106.017	104.622
T_age_min	D.F.	1	
	χ^2	64.624	
Q_codesnippet_num	D.F.	1	
	χ^2	50.250	
B_solved_likelihood_max	D.F.	3	2
	χ^2	44.900	41.039
Q_body_len	D.F.	1	
	χ^2	29.932	
T_age_max	D.F.	1	
	χ^2	21.996	
Q_url_num	D.F.	1	
	χ^2	17.373	
U_asker_answer_num	D.F.	1	
	χ^2	12.798	

5.2 Results

Our model explains our dataset well and has a reliable performance. Table 4 shows the result of the performance analysis of our model. Our model obtains a median AUC of 0.708, which indicates that the model explains the relationship between the studied factors and the solving-likelihood well. In addition, the low optimism of the AUC value (i.e., 0.001) suggests that our model does not suffer from overfitting.

A question that received more answers before a bounty was proposed has a higher solving-likelihood, especially when the question has more than 3 answers before the proposal of the bounty. Table 4 shows the Wald's χ^2 value of the studied factors. The *Q_answer_num* factor (i.e., the number of answers that a question received before a bounty is proposed) contributes the most explanatory power to the model. Figure 10 shows the relationship between the bounty question solving-likelihood and *Q_answer_num*. *Q_answer_num* has a positive relationship with the solving-likelihood. Once a question has more than three answers, the solving-likelihood of the question is at least 0.9, while the solving-likelihood for questions without an answer is 0.59. One possible explanation is that answerers may benefit from the prior answers of a question. The more prior answers the question has, the more potential solvers are likely to benefit from those answers. For example, the poster of the accepted answer to a question[15] mentioned that "The answer by Yacoby can be extended further." In other words, the accepted answer was based on a prior answer.

The bounty value and the timing of proposing a bounty are important factors that are associated with the solving-likelihood of a bounty question. Table 4 shows that *B_value* (i.e., the bounty value) and *B_days_before_bounty* are the second and fourth most important factors in the model. In Figure 10, we observed a positive relationship between the bounty value and the bounty question solving-likelihood. One possible explanation is that higher bounties attract more attention to a question, thereby increasing the solving-likelihood.

Figure 10 also shows a negative relationship between *B_days_before_bounty* and the solving-likelihood of a bounty question, which indicates that a question for which a bounty is proposed earlier may have a higher likelihood of being solved. We also noticed that after 365 days, the bounty question solving-likelihood drops drastically. We suggest bounty backers to consider proposing bounties earlier. In Section 7, we further study the interesting relationship between the timing of a bounty and the attention (or *traffic*) that it draws to a question.

The associated communities of a bounty question have a significant relationship with its solving-likelihood. The solving likelihood of a tag for non-bounty questions reflects the question-solving skill level of answerers in the community of that tag. Table 4 shows that *T_solving_likelihood_nor-*

[15] https://stackoverflow.com/questions/1809670/how-to-implement-serialization-in-c

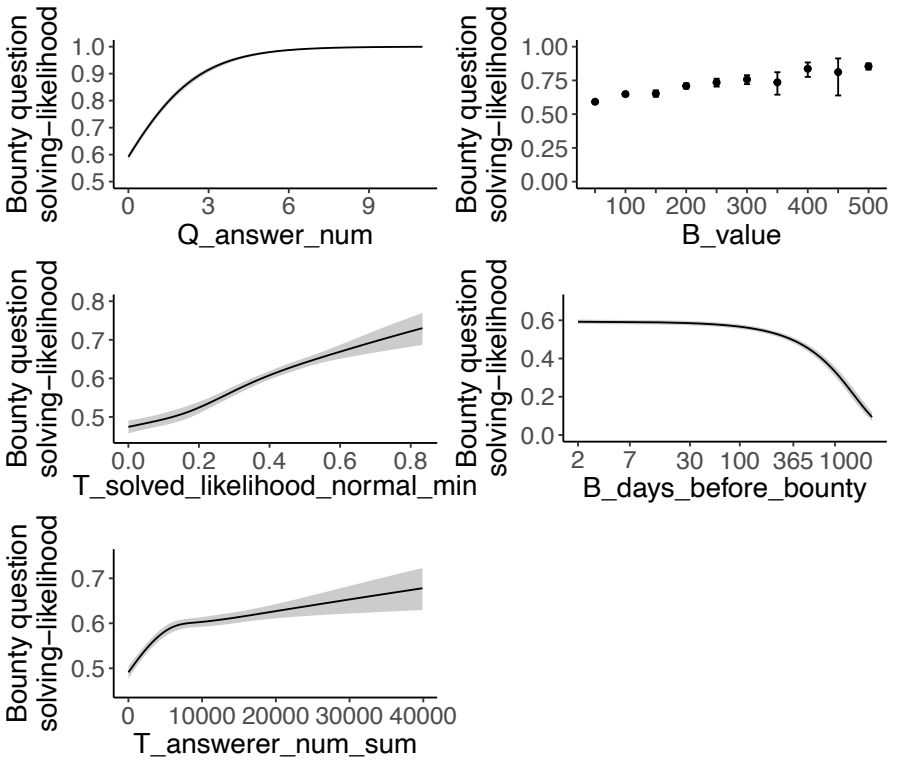


Fig. 10: The relationship between the five most important factors and the bounty question solving-likelihood in the logistic regression model. For each plot, we set all the factors except the studied factor to their median value in the model while varying the studied factor. The grey area represents the 95% confidence interval. The B_value uses a dot plot instead of a line plot because it is an ordinal variable, as B_value is between 50 and 500 (with an interval of 50), while the other variables are natural numbers.

mal_min plays the third most important role in the model which means that the lowest question-solving skill level of answerers in the associated communities of a bounty question has a significant impact on the solving-likelihood of the bounty question. Table 4 also shows that *T_answerer_num_sum* (i.e., the total answerer population of the associated communities of a bounty question) plays the fifth most important role in the model. In other words, the number and question solving-likelihood of the answerers in the communities in which a bounty question is asked have an important impact on the solving-likelihood of a question.

In addition, Figure 10 shows that *T_solving_likelihood_normal_min* and *T_answerer_num_sum* both have a positive relationship with the solving-likelihood of a bounty question. Hence, bounty questions that are asked in small commu-

nities, or communities in which the answerers have a relatively low question solving-likelihood, are less likely to be solved.

5.3 The Important Factors for the Solving-likelihood of Non-bounty and Bounty questions

To further understand the important factors for solving-likelihood of non-bounty and bounty questions, we built two additional models to explain the important factors of the solving-likelihood of non-bounty (*non-bounty-question-model*) and bounty questions (*bounty-question-model*$_{without\_bounty\_factors}$). To be able to compare the factors, we used only non-bounty-related factors in these two models. Table 5a and Table 5b show the performance and the top five most important factors for the

We found that *T_solving_likelihood_normal_min* and *T_solving_likelihood_normal_max* are the most important factors for both models, which indicates that the question solving-likelihood of a tag is important for both bounty and non-bounty questions. Aside from the factors in the tag dimension, the factors in the question dimension are important for the non-bounty questions (e.g., the length of the question body, the number and the proportion of code snippets in a question). In contrast, for the bounty questions all top five most important factors are tag related.

Table 5: The result of our logistic regression models for understanding the relationship between the non-bounty factors and the solving-likelihood of two types of questions (i.e., bounty and non-bounty questions). The factors are ordered by their importance (i.e., overall Wald's χ^2 value) in the model. We only show the top five factors which contribute the most significant importance (i.e., the p-value is less than 0.002) to our models.

(a) *Non-bounty-question-model*

Factors	**Median value**		
AUC	0.668		
AUC optimism	0.001		
Factors		**Overall**	**NL**
T_solving_likelihood_-normal_min	D.F.	4	3
	χ^2	511.513	38.462
T_solving_likelihood_-normal_max	D.F.	3	2
	χ^2	325.337	34.475
Q_body_len	D.F.	1	
	χ^2	148.600	
Q_codesnippet_num	D.F.	1	
	χ^2	126.252	
Q_codesnippet_-proportion	D.F.	1	
	χ^2	120.665	

(b) *Bounty-question-model*$_{without\_bounty\_factors}$

Factors	**Median value**		
AUC	0.670		
AUC optimism	0.001		
Factors		**Overall**	**NL**
T_solving_likelihood_-normal_min	D.F.	4	3
	χ^2	979.604	89.328
T_solving_likelihood_-normal_max	D.F.	3	3
	χ^2	864.212	89.213
T_answerer_num_sum	D.F.	4	3
	χ^2	715.245	89.213
T_age_min	D.F.	1	
	χ^2	199.427	
T_age_max	D.F.	1	
	χ^2	136.787	

Table 6: The 5-number summaries for the solving-times of the fast-solved and slow-solved bounty questions.

Question Type	Quantile solving-time (days)				
	Min	1^{st}	Median	3^{rd}	Max
Fast-solved	0.00	0.02	0.04	0.08	0.14
Slow-solved	4.60	5.35	6.07	6.67	8.06

The number of answers before the proposal of a bounty and the value of a bounty are the most important factors that impact the solving-likelihood of a bounty question. In addition, the solving-likelihood of bounty questions is higher in larger communities where the question solving-likelihood of answerers is higher.

6 What Are the Important Factors that Are Associated with the Solving-time of a Bounty Question?

In Section 4, we observed that the solving-time of bounty questions varies across tags while the bounty value has no relation with the solving-time. In this section, we study which other factors are related to the solving-time of a bounty question. With a better understanding of this relationship, we can provide insights into how to use a bounty to speed up the process of getting a bounty question solved.

6.1 Approach

Similar to Section 5, we built a logistic regression model to study the relationship between the studied factors and the likelihood of a bounty question being solved fast. Similar to prior studies (Wang *et al.*, 2018c; Tian *et al.*, 2015), we sorted the solved bounty questions by their solving-time (in days) in ascending order and labeled the top 20% questions as fast-solved bounty questions, and the bottom 20% as the slow-solved bounty questions. Table 6 shows the 5-number summaries for the solving-times for fast-solved and slow-solved bounty questions. In the following subsections, we explain the additional studied factors compared to solving-likelihood model and the model construction. We analyzed our model for the solving-time in the same way as discussed in Section 5.

6.1.1 Additional Studied Factors

We studied 8 factors in the user dimension in addition to the 34 factors that we included in our solving-likelihood model in Section 5. These eight factors (i.e.,

Table 7: The description of and rationale for the additional factors that we studied in our logistic regression model for the likelihood of a bounty question being solved fast. The factors marked with '**' are the time-dependent factors which are calculated considering only the activity within a month before the bounty was offered.

Factor name	Description	Rationale
User		
U_answerer_answer_num **	The number of answers that the answerer posted previously.	A previously active answerer may answer questions faster.
U_answerer_question_num **	The number of questions that the answerer posted previously.	
U_answerer_question_score_max/median/sum	The max/median/sum scores of the answerer's prior questions.	These factors indicate the question solving and asking-skills of an answerer and may influence the solving-time of a question.
U_answerer_answer_score_max/median/sum	The max/median/sum scores of the answerer's prior answers.	

Table 7) reflect the activity and the question solving-likelihood of answerers whose answers were awarded with bounties. These eight new factors are not included in the model in Section 5 since they are related to the answer and answerer of a question, which would not be available for the unsolved bounty questions that we studied in Section 5. Hence, our model for the solving-time contains 42 factors in total. Also note that we cannot include unsolved questions in our model since such questions would not have a solving-time.

6.1.2 Model Construction

We applied the same correlation and redundancy analysis for these 42 factors as discussed in Section 5 to remove correlated and redundant factors. Finally, we ended up with seven factors in the question dimension, seven factors in the user dimension, five factors in the bounty dimension and seven factors in the tags dimension (Figure 11). We also used the same approach as in Section 5 to add degrees of freedom to non-linear factors.

6.2 Results

Our model explains our dataset well and has a reliable performance. Table 8 shows the results of the performance analysis of our model. Our model has a high median AUC of 0.817 which indicates that the model explains the relationship between the studied factors and the likelihood of being solved fast well. In addition, the low optimism of the AUC values (i.e., 0.002) suggests that our model does not overfit the dataset.

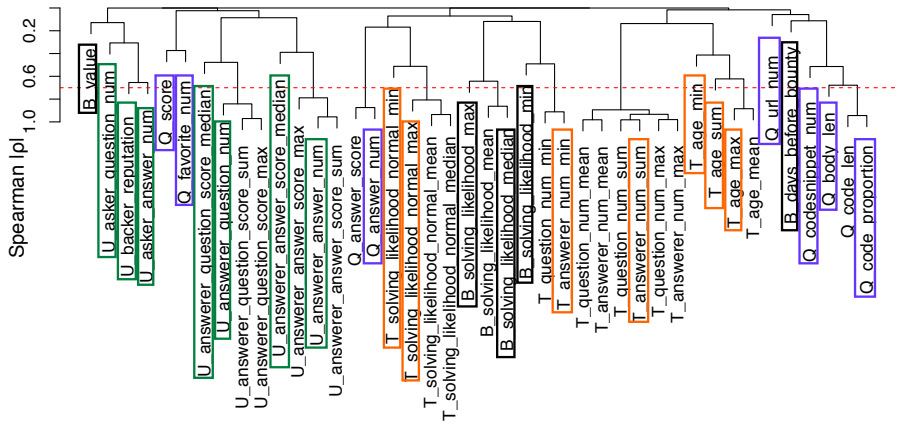


Fig. 11: The hierarchical clustering plot of factors in our solving-time model. According to the Spearman rank correlation test (using a cut-off value of 0.7), we selected the simplest metrics to compute across each dimension of correlated factors. We ended up with seven factors in the question level dimension (marked in blue), seven factors in the user dimension (marked in green), five factors in the bounty dimension (marked in black), and seven in the tags dimension (marked in orange).

The number of answers that a question received before a bounty was proposed (i.e., *Q_answer_num*) is the most important factor to get a bounty question solved fast. Table 4 shows the Wald's χ^2 value of the studied factors. Similar to the solving-likelihood model in Section 5, *Q_answer_num* contributes the most explanatory power to the solving-time model. Figure 12 shows the relationship between *Q_answer_num* and the likelihood of solving a bounty question fast. The more answers that a question received before a bounty was proposed, the faster it was solved. In addition, the likelihood of a bounty question getting solved fast increases sharply as the number of previously posted answers goes from zero to three answers. After three answers, the likelihood of getting solved fast remains equally high. A possible explanation is similar to the one in Section 5. More answers may contain useful information that help other answerers to provide an acceptable answer to the question.

The activity level of the answerer has a positive relationship with the likelihood of solving a bounty question fast. *U_answerer_answer_num* is the second most important factor in the model and has a positive relationship with the likelihood of getting a bounty question solved fast (see Figure 12). This finding is similar to what we observed in our prior work (Wang *et al.*, 2018c), which is that the activity level of answerers is the most important factor that impacts the speed of a question getting solved on Stack Overflow.

Higher bounty values are not associated with faster solving of questions. The value of a bounty (i.e., *B_value*) is of low importance in our

Table 8: The result of our logistic regression model for understanding the relationship between the studied factors and the likelihood of a bounty question being solved fast. The factors are ordered by their importance (i.e., overall Wald's χ^2 value) in the model. We also show the non-linear (NL) Wald χ^2 value. We only show factors which are of significant importance (i.e., the p-value of the χ^2 is less than 0.002 (i.e., 0.05/26)) in our model. See our supplementary material (Zhou, 2019) for the full table.

Factors		**Solving-time Model**	
AUC		0.817	
AUC optimism		0.002	
Factors		**Overall**	**NL**
Q_answer_num	D.F.	1	
	χ^2	2032.150	
U_answerer_answer_num	D.F.	3	2
	χ^2	581.880	361.452
T_solving_likelihood_normal_min	D.F.	3	2
	χ^2	391.171	76.639
T_age_max	D.F.	1	
	χ^2	317.732	
T_solving_likelihood_normal_max	D.F.	1	
	χ^2	243.640	
B_days_before_bounty	D.F.	1	
	χ^2	173.308	
Q_code_proportion	D.F.	1	
	χ^2	144.062	
Q_favorite_num	D.F.	1	
	χ^2	74.265	
Q_body_len	D.F.	2	
	χ^2	58.913	
T_age_sum	D.F.	1	
	χ^2	45.573	
T_answerer_num_sum	D.F.	1	
	χ^2	42.458	
Q_codesnippet_num	D.F.	1	
	χ^2	15.294	
B_solving_likelihood_max	D.F.	1	
	χ^2	14.696	

model. This might be due to various reasons. For instance, it might take longer to solve high-valued bounty questions due to them being harder or less popular.

The question solving-likelihood of associated communities have a significant impact on the likelihood of solving a bounty question fast. The lowest question-solving skill level of the associated communities (i.e., *T_solving_likelihood_normal_min*) plays the third important role in the model. The *T_solving_likelihood_normal_min* has a positive relationship with the likelihood of a bounty question getting solved fast.

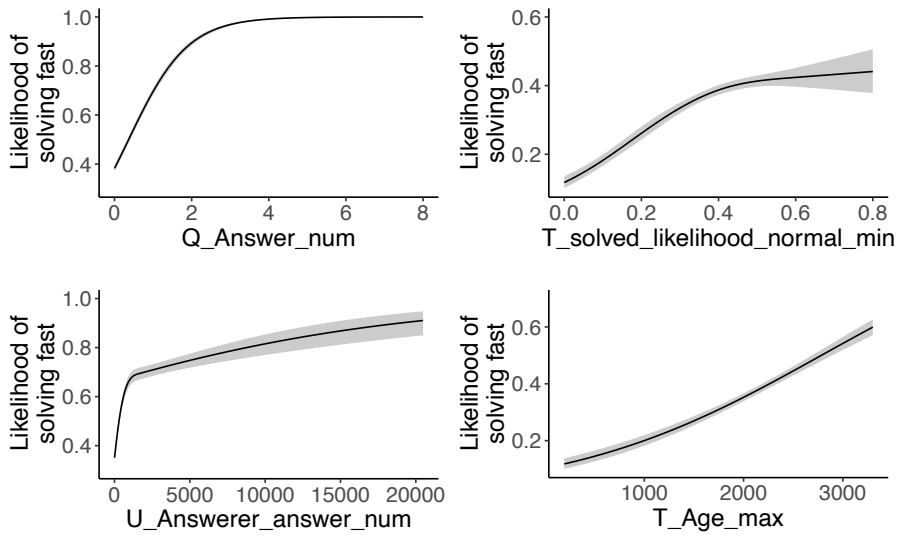


Fig. 12: The relationship between the studied factors and the likelihood of a bounty question getting solved fast in the logistic regression model. For each plot, we set all the factors except the studied factor to their median value in the model while varying the studied factor. The grey area represents the 95% confidence interval.

The number of (unaccepted) answers to a question before a bounty is proposed has the strongest association with the likelihood of a bounty question solving fast. A higher-valued bounty does not help a bounty question to get solved faster. The activity level of potential answerers and the question solving-likelihood of the potential answerer communities have a strong association with the solving-time of a bounty question.

7 Studying the Association between Bounties and the Traffic of Questions

In the previous sections, we found that the popularity of a question (e.g., in terms of the number of existing answers) and the size and question solving-likelihood of the community in which the question is asked (e.g., in terms of the solving-likelihood of questions with a certain tag) are strongly associated with the solving-likelihood and solving-time of a question. These findings suggest that it is beneficial to attract more traffic to a question. In this section, we conduct an empirical study of the association between a bounty and the traffic to the bounty question.

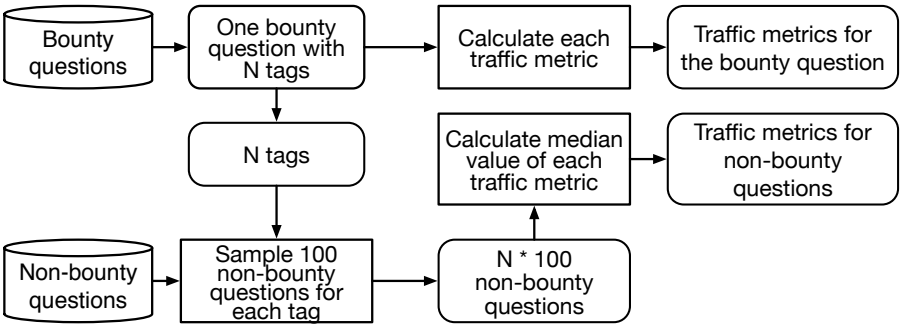


Fig. 13: An overview of our approach for computing the traffic of bounty and non-bounty questions.

7.1 Approach

A bounty in Stack Overflow will be active for a maximum of seven days. Therefore, we only measure the traffic for seven days after the bounty was proposed. We use the following metrics to capture the traffic of a question:

1. **The number of new answers** to a question.
2. **The number of new comments** on a question.
3. **The number of new edits** of answers to a question.

To understand how bounties impact the traffic to a question, we compared the traffic between bounty and non-bounty questions. If the traffic of bounty questions is significantly higher than that of non-bounty questions, it suggests that bounties may help to attract more traffic to a question.

The traffic of a question could be impacted by several conditions under which the question was asked, such as the content of the question, the related tags, and the creation time of the question. Ideally, we can compare bounty and non-bounty questions that share the same conditions. However, due to the richness of the metadata of the questions on Stack Overflow, it is very hard to find a perfect match for the bounty questions automatically. Therefore, we use a sample-based method to identify a set of questions that share similar conditions as a bounty question, and we use the median value of their traffic metrics to represent the traffic of non-bounty questions.

Figure 13 gives an overview of our approach for calculating the traffic metrics of bounty and non-bounty questions. The details of each step are explained below.

1. For each bounty question, we calculate its seven-day traffic metrics since the bounty was proposed. In other words, if the bounty was proposed on the m^{th} day after the creation of the question, we calculate the traffic from day m to $m + 7$.
2. We extract all N tags of the bounty question. For each tag, we randomly sample 100 questions from the non-bounty questions that are associated

with the tag (without repetition). After this step, we have $N * 100$ sampled non-bounty questions.
3. For each sampled non-bounty question, we calculate the seven-day traffic between m and $m + 7$ in the same way as we do for the bounty question.
4. We use the median value of each traffic metric of the sampled non-bounty questions to represent the traffic of the corresponding non-bounty questions.

After calculating the traffic metrics for all bounty questions and their similar non-bounty questions, we categorized them into groups based on the *days-before-bounty* metric to study the impact of this metric on the traffic as prior studies (Hanrahan *et al.*, 2012; Wang *et al.*, 2018c). We define the *time-based groups* as follows:

1. **[3, 3]:** the bounty is proposed on the third day after the question was created (i.e., the earliest allowed by Stack Overflow – see Section 2.2).
2. **[4, 7]:** the bounty is proposed at least four and at most seven days after the question was created.
3. **[8, 30]:** the bounty is proposed at least 8 and at most 30 days after the question was created.
4. **[31, 365]:** the bounty is proposed at least 31 and at most 365 days after the question was created.
5. **[366,∞):** the bounty is proposed at least 365 days after the question was created.

We then compared the traffic between bounty and non-bounty questions as described above across these groups.

To study the impact of the bounty value on traffic, we compared the traffic metrics of bounty questions across different bounty values. We categorized bounty questions into three groups based on their bounty value as identified in Section 4. The *bounty-value-based groups* are as follows:

1. **Small (bounty):** the question has a bounty value that ranges from 50 to 150.
2. **Moderate (bounty):** the question has a bounty value that ranges from 200 to 350.
3. **Large (bounty):** the question has a bounty value that ranges from 400 to 500.

To compare the differences of the traffic metrics between bounty and non-bounty questions, we used the Wilcoxon rank-sum test and Cliff's delta effect size as explained in Section 4.1.

7.2 Results

Questions are likely to attract more traffic than non-bounty questions after they receive a bounty. Figure 14 shows the seven-day traffic

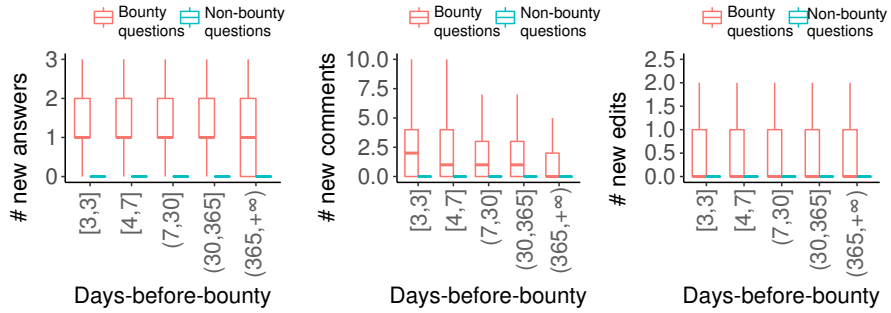


Fig. 14: The distributions of the traffic metrics (i.e., the number of new answers, new comments and new edits) for bounty and non-bounty questions across different values of the days-before-bounty metric.

for bounty and non-bounty questions. For the same time-based group, the traffic of bounty questions is always higher than that of non-bounty questions. For each time-based group, the statistical tests show that the differences in the traffic between bounty and non-bounty questions are significant with large effect sizes. The results indicate that bounty helps attract new traffic. For example, the question[16] was created on Nov 27, 2009, and received five answers before the bounty was proposed on Jan 28, 2015. After proposing the bounty, 12 new answers were created. An interesting additional observation is that non-bounty questions receive hardly any traffic after 2 days. This observation is a confirmation of the finding in prior work that a question that is not solved fast, is unlikely to be solved at all (Anderson *et al.*, 2012).

To ensure that the above finding is not biased by the popularity of bounty questions (e.g., bounty questions may attract more traffic in general compared to non-bounty questions), we also calculated the (absolute) difference in traffic to a question before and after proposing the bounty. To calculate this difference, we subtracted the value of the traffic metrics before proposing the bounty from the seven-day traffic values after proposing the bounty. Figure 15 shows the distributions of these differences. Figure 15 shows that the median difference is always at least zero, and in most groups larger than zero for the number of new answers and new comments. These differences indicate that the traffic to a bounty question increased in most cases after the proposal of a bounty.

Questions with bounties that are proposed early are more likely to have more comments than questions with bounties that are proposed later. Figure 14 shows the distribution of traffic metrics across the time-based groups. We can observe that proposing a bounty earlier is not correlated with a higher number of new answers and edits, but it is correlated with a higher number of new comments. We used the Wilcoxon rank-sum test

[16] https://stackoverflow.com/questions/1809986/

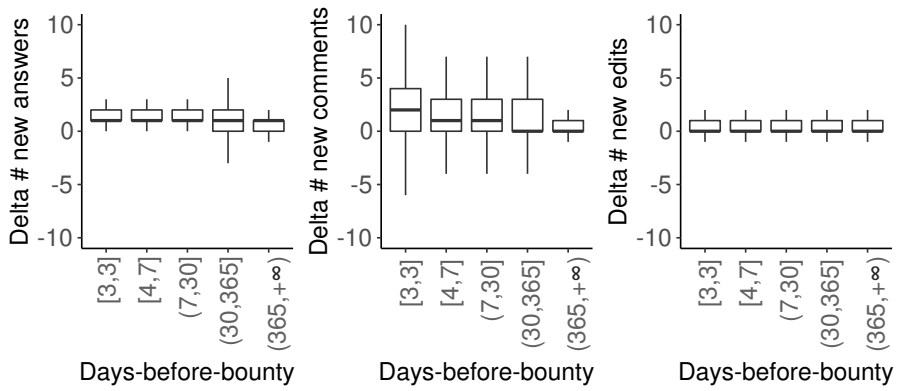


Fig. 15: The distributions of the (absolute) difference in traffic to a question before and after proposing a bounty. The difference (delta) metrics are calculated by subtracting the value of a traffic metric before the bounty was proposed from the seven-day traffic metric value (i.e., after - before).

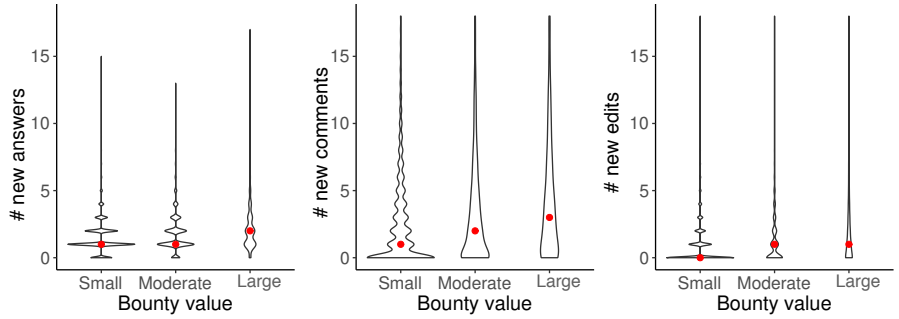


Fig. 16: The distributions of the traffic metrics (i.e., the number of new answers, new comments and new edits) for bounty and non-bounty questions across different bounty value groups. The red dot is the median value of the corresponding distribution.

and Cliff's delta d to measure the differences in the traffic metrics between each pair of adjacent time-based groups. We also performed a Bonferroni correction (Bonferroni, 1936) to correct the p-values for multiple comparisons. All pairwise comparisons show that the differences are significant (i.e., the p-value $< 0.05/4$) with a non-negligible effect size in terms of the number of comments, suggesting that the number of comments is positively correlated with the timing of proposing a bounty.

A higher-valued bounty is more likely to attract more traffic to a question, especially when the bounty value is over 400. Figure 16 shows the distributions of the traffic metrics of bounty questions across the bounty-value-based groups. We observed that a question with a higher-valued

bounty is more likely to attract more traffic. More specifically, a question with a large bounty (i.e., with a bounty value of at least 400) attracts more answers than ones with a small bounty (i.e., with a bounty value of 150 or less). Similar trends hold for the number of new comments and edits. Our statistical test results show that the differences between each pair of adjacent bounty-value-based groups are significant. Moreover, the effect size of the differences between the small bounty and large bounty groups are at least small for all the traffic metrics, indicating that a large bounty is more likely to attract additional traffic to a question.

Questions are likely to attract more traffic after receiving a bounty than non-bounty questions, particularly for questions that receive a bounty with a value of at least 400.

8 Further Analysis on Bounties and a Discussion on the Implications of our Findings

In this section, we discuss bounties for rewarding existing answers, and we look into the differences between unsolved and solved bounty questions. We also discuss the implications of our findings.

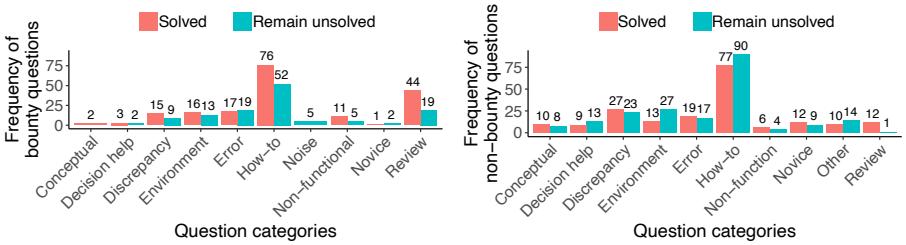
8.1 Bounties for Rewarding Existing Answers

3% of the bounties were proposed to reward an existing answer. In addition to the main purpose of getting a question solved, we observed (from the user-posted reason for the bounty) that 3,894 out of 129,202 (3%) bounties were proposed to reward an existing answer. We refer to this type of bounty as a *bonus bounty*. The median answer score (i.e., the number of upvotes from users) of the answers that were awarded a bonus bounty is 8, while the median score for the other answers, and for accepted answers on Stack Overflow is only 1. In other words, the rewarded existing answers appear to be of a higher quality than the average answer on Stack Overflow.

Bounty backers who proposed bonus bounties are usually users with a high reputation. The median number of reputation points of the bounty backers who proposed a bonus bounty is 4,570, which is six times higher than the reputation points of other bounty backers (i.e., 706). Such backers are usually much "richer" (i.e., have a larger amount of reputation points) than other backers. Moreover, bonus bounties tend to be larger than non-bonus bounties. While the median value is 50 for both types of bounties, the mean value of a bonus bounty is 113 while the mean value of a non-bonus bounty is 82, which indicates that bonus bounties tend to have a higher value. Finally, 55% of the backers of bonus bounties are not the asker of the bounty question (vs. only 15.7% for non-bonus bounties).

Table 9: The question categories and examples as defined by Treude *et al.* (2011). Note: this table is reprinted from Treude *et al.* (2011).

Name	**Definition**	**Example**
How-to	Questions that ask for instructions.	*How to crop image by 160 degrees from center in asp.net?*
Discrepancy	Some unexpected behavior that the person asking the question wants explaining.	*getElementById() returns null even though the element exists?*
Environment	Questions about the environment either during development or after deployment.	*Setting Environment Variables in Rails 3 (Devise + Omniauth)?*
Error	Questions that include a specific error message.	*Getting an ambiguous redirect error.*
Decision help	Asking for an opinion.	*Should I use JSLint or JSHint JavaScript validation?*
Conceptual	Questions that are abstract and do not have a concrete use case.	*Content-Disposition: What are the differences between "inline" and "attachment"?*
Review	Questions that are either implicitly or explicitly asking for a code review.	*Is my file struts.xml is it correct?*
Non-functional	Questions about non-functional requirements such as performance or memory usage.	*Where to store global constants in an iOS application?*
Novice	Often explicitly states that the person asking the question is a novice.	*How to use WPF Background Worker?*
Noise	Questions not related to programming.	*Apple Developer Program.*
Other	Questions that are other than the above categories.	*Where do I find old versions of Android NDK?*



(a) (b)

Fig. 17: The frequency of categories of our samples bounty questions (a) and (b) non-bounty questions from a prior study (Treude *et al.*, 2011).

8.2 Remain-Unsolved vs. Solved Bounty Questions

There are 44,635 bounty questions of which the bounties expired with no awarded answers. We observed that 64.4% (28,754) of those questions were

never solved after that (i.e., *remain-unsolved bounty questions*). To compare the differences between remain-unsolved and solved bounty questions, we sampled 100 (out of 28,754) bounty questions which had no accepted answer at the time of collecting our data, and 100 (out of 79,093) bounty questions which were solved as the two statistically representative samples with a 95% confidence level and a 10% confidence interval. The first two authors manually and independently labeled the categories of these 200 bounty questions into the categories that were defined by Treude *et al.* (2011) (see Table 9). They discussed conflicts until a consensus was reached. The Cohen's kappa (Gwet *et al.*, 2002) value to measure the inter-rater agreement of this labeling is 0.66 before resolving the conflicts. Note that in our study, some questions have multiple categories. When we calculated the Cohen's Kappa, we did not consider partial agreements, instead we consider the labeled categories of a question from two raters in agreement only when their categories were exactly the same. For example, if a question was assigned categories c1 and c2 by rater 1 and categories c2 and c3 by rater 2, we considered them in disagreement. If we considered partial agreements, the Cohen's Kappa would be 0.95.

Figures 17a and 17b show the frequency of question categories for our sampled bounty and non-bounty questions which were studied by Treude *et al.* (2011). We observed that the category "How-to" is the most popular category for both bounty and non-bounty questions. However, the solving-rate of the "How-to" category for bounty questions is 59%, which is higher than that of non-bounty questions (46%). We also observed that bounty questions in the "Review" category are more likely to be solved with a solving-rate of 70% (i.e., 44 out of 63) for bounty questions and 92% (i.e., 12 out of 13) for non-bounty questions. One possible explanation is that review questions may be easier to solve as there is more information about the problematic source code in the question. For example, one question about Biztalk[17] provides a clear description, development environment and code snippet, which makes it easier for answerers to solve the question. Moreover, we observed that "Review" questions are more likely to appear in bounty questions (32%) than non-bounty questions(3%).

8.3 The Implications of Our Findings

While bounties are not a silver bullet for getting a question solved, bounty questions tend to have a higher solving-likelihood than non-bounty questions, particularly when focusing on long-standing unsolved questions. As we showed in Section 7, in general bounties attract more traffic to questions. In addition, the solving-likelihood of bounty questions is higher than that of non-bounty questions, particularly for long-standing unsolved questions (see Section 4). For example, the solving-likelihood of questions that were unsolved for 100 days increases from 1.7% to 55% after proposing a bounty.

[17] http://bit.ly/2HsnxbY

The sweet spot for proposing a bounty is as soon as Stack Overflow allows it. Stack Overflow does not allow the proposal of a bounty within two days after the posting of a question. We observed in Section 7 that after these two days, the traffic to the vast majority of questions is negligible. Hence, we recommend that in order to maximize the solving-likelihood of a question, the bounty is best proposed as soon as possible after those two days. Section 5 confirms that the solving-likelihood is the highest for bounties that are proposed after two days.

Stack Overflow should indicate which communities (tags) are more active and have a higher solving-likelihood of bounty questions. We showed in Sections 5 and 6 that the number of prior answers (i.e., *Q_answer_num*) is the most important factor for both the solving-likelihood and the solving-time of a bounty question. In addition, in these sections, we observed that the size and question solving-likelihood of a community are important factors when it comes to the solving-likelihood of a bounty question. Stack Overflow should provide guidance to bounty backers about which communities are most likely to benefit from proposing a bounty.

Bounty backers should be aware that a highly-valued bounty increases the solving-likelihood of a question, but does not guarantee a fast answer. Sections 5 and 7 show that a higher bounty value attracts more traffic to and increases the likelihood of a question. However, Section 6 shows that the bounty value contributes little to speed up the solving of a question. We recommend that Stack Overflow provides its users with an estimate of the solving-likelihood and solving-time when proposing a bounty. These estimates can be retrieved from historical data about the success of bounties in a particular community, similar to the analysis that we conducted in this paper.

9 Threats to Validity

In this section, we discuss the threats to validity. Threats to **external validity** are related to the generalizability of our findings. We studied only bounty questions on Stack Overflow. Further research should investigate whether our findings are generalizable to other Q&A websites, including non-technical ones (such as the other Stack Exchange websites). In addition, although our models have high explanatory power, there might be additional factors that relate to the solving-likelihood and solving-time of bounty questions. Future studies should explore additional factors.

Threats to **internal validity** relate to the experimenter bias and errors. One threat is that we rely on manual analysis to categorize the questions in Section 8, which may introduce a bias due to human factors. To mitigate the threat of bias during the manual analysis, two of the authors conducted the manual analysis. We also measure the inter-rater agreement using Cohen's kappa and the raters discussed their differences until they reached consensus. While this manual analysis is only a small part of our study, future studies

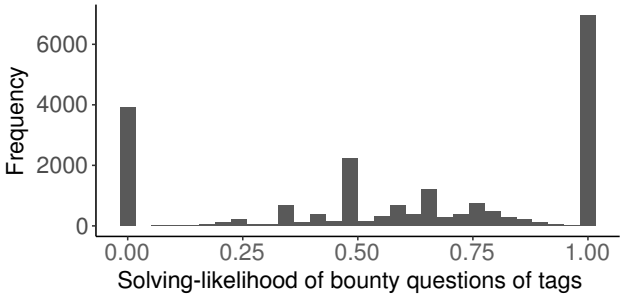


Fig. 18: The distribution of the solving-likelihood of tags of bounty questions without filtering tags.

should investigate how questions can be classified automatically to reduce the human classification bias and error.

One threat to the internal validity of our study is our categorization of the fast-solved bounty questions (i.e., the fastest 20%) and slow-solved bounty questions (i.e., the slowest 20%) in Section 6. We conducted a sensitivity analysis by building the logistic regression model using different thresholds (i.e., 30% and 40%) for slow and fast-solved questions. The built models still had high median AUC and low median AUC optimism values (i.e., 0.78 and 0.002 for the 30% threshold, and 0.74 and 0.002 for the 40% threshold). Moreover, the top four important factors were consistent with the model that was built using the 20% threshold. Therefore, we can conclude that our observations are not particularly sensitive to the threshold that we selected to distinguish slow and fast-solved question.

We selected five as the threshold to filter tags in Section 4, which is a threat to the internal validity of our study. Figure 18 shows the distribution of the solving-likelihood of different tags of bounty questions without filtering tags. Many of the extreme values (0 and 1) are not meaningful due to the very low number of questions in those tags. We agree that five is an arbitrary threshold, unfortunately, any other threshold will be arbitrary as well but we feel it is necessary to put one to enable a clearer representation of the results.

The way that we selected the non-bounty questions for traffic analysis in Section 7 is a threat to internal validity. We considered the tags of questions, while there may be other confounding factors that could impact the traffic of questions. Future studies should investigate other techniques for matching bounty questions to non-bounty questions.

One threat to the internal validity of our study is that we measured only the traffic within seven days of posting the bounty. Hence, we did not take any long-term effects of the bounty into account. The main reason is that it is not possible to decide whether these long-term effects were likely caused by the bounty, or by something else. While we cannot claim this causality for the seven-day traffic either, it is more likely that the bounty has a relationship with an increase in traffic while it is active.

A final threat to the internal validity of our study is that while we studied various confounding factors across several dimensions (i.e., the question, user, bounty, tag, answer, and answerer dimensions), there may exist other factors that might potentially have an impact on the solving-likelihood and solving-time of bounty questions. Future studies should investigate the impact of other factors.

10 Related Work

In this section, we discuss prior work that is related to our study. We focus on prior work in two research areas: (1) understanding incentive systems in a software engineering context and (2) improving the question answering process on Stack Overflow.

10.1 Understanding Incentive Systems in a Software Engineering context

Non-monetary incentive systems: A number of studies examined the non-monetary incentive system of Q&A sites, which often consists of one or more gamification element(s). Anderson *et al.* (2013) found that a badge can increase the overall level of user participation on the site. The extent to which the participation is increased depends on how close the user is to the badge boundary. Cavusoglu *et al.* (2015) also performed an empirical study on how gamification on Stack Overflow stimulates voluntary participation. Nakasai *et al.* (2018) found that donation badges can reduce developers' response time on bug reports. These observations are similar to our observation that a bounty can help attract traffic to a question. Beside the user participation which was investigated by prior studies, we also investigate the impact of bounties on the solving-likelihood and solving-time of a question.

Monetary incentive systems: Monetary incentives (often called *bounties*) are used to motivate developers to complete software engineering tasks. Prior work has studied the impact of bounties on various software engineering tasks (e.g., bug fixing and finding security vulnerabilities). In our prior work (Zhou *et al.*, 2019), we studied bounties from the BountySource platform. We found that bounty backers risk financial loss if they invest money by proposing bounties on long-standing issue reports. Krishnamurthy and Tripathi (2006) gave an overview of bounties in Free/Libre/Open Source Software (FLOSS). They observed that bounty hunters' responses are related to the workload, the probability of winning the bounty, and the value of the bounty. Zhao *et al.* (2014) investigated the characteristics of hunters in bug-bounty programs and found that the diversity of hunters increased the productivity of the vulnerability discovery process. Kanda *et al.* (2017) studied Bountysource to understand the association between bounties and the issue-fixing process in open source projects. Finifter *et al.* (2013) studied vulnerability rewards programs for Chrome and Firefox and found that the rewards programs for both projects are economically effective, compared to the cost of hiring full-time security researchers.

Munaiah and Meneely (2016) analyzed the relationship between the Common Vulnerability Scoring System (CVSS) scores and the awarded bounty vulnerabilities and found a weak negative correlation between CVSS scores and bounties. Hata *et al.* (2017) studied the heterogeneity of bug bounty program contributors. Zhao *et al.* (2017); Maillart *et al.* (2017) analyzed the effect of different policies of bug-bounty programs. By studying bug-bounties from several perspectives, they provided insights on how to improve the bug-bounty programs. For example, Maillart *et al.* (2017) suggested project managers to dynamically adjust the value of rewards according to the market situation (e.g., by increasing the rewards when releasing a new version).

The above prior studies focus on monetary incentives for a variety of software engineering tasks. Different from these studies, we focus on non-monetary incentives which are employed in a popular technical Q&A site (Stack Overflow). We found that a higher bounty value increases the likelihood of a question getting solved, but does not expedite the solving process of a question.

In addition, several studies explored the impact of the monetary incentive systems which are available on some (non-technical) Q&A sites. Hsieh *et al.* (2010) performed an empirical study on questions of a pay-for answer site (i.e., Mahalo Answers) and found that askers are more likely to pay when requesting facts and will pay more when questions are more difficult. They also found that questions with higher rewards have a higher archival value. Jan *et al.* (2017) analyzed the benefit and potential concerns of the monetary incentive system on two Q&A sites. For example, they showed that monetary incentive systems can improve the speed of getting answers and the quality of questions.

10.2 Improving the Question Answering Process on Stack Overflow

Nowadays, developers rely heavily on Stack Overflow to help solve many software engineering problems. Therefore, it is important to understand the question answering process on Stack Overflow, so that potential improvements can be identified to benefit the users of Stack Overflow. Many prior studies were done in this direction. Wang *et al.* (2018c) used logistic regression models to study the impact of factors along four dimensions (i.e, answers, questions, askers, answerers) on the speed of a question getting an accepted answer on Stack Overflow and three other famous Q&A Stack Exchange websites. They found that non-frequent answerers are the bottleneck for fast answers and they suggested that Stack Overflow should consider improving their incentive system to motivate non-frequent answerers. Our findings also echo that the answerers of a tag are important for both the solving likelihood and solving time of a bounty question that is associated with that tag. In order to help users to find the right channel to ask questions, several approaches have been developed to help users generate tags automatically when they post a question (Wang *et al.*, 2018a; Xia *et al.*, 2013; Liu *et al.*, 2018; Wang *et al.*, 2014).

To improve the quality of answers on Stack Overflow, Ponzanelli *et al.* (2014) proposed an approach to identify low-quality questions. Srba and Bielikova

(2016) evaluated how low-quality content on Stack Overflow negatively impacts the community, and proposed ways to solve the problem. Chen *et al.* (2018) developed a convolutional neural network-based approach to learn editing patterns from historical post edits for predicting the need for editing a post. They also developed an approach that recommends editorial suggestions to improve the quality of a post (Chen *et al.*, 2017). Wang *et al.* (2018b) analyzed how users revise answers on Stack Overflow under the current badge system and provided suggestions to improve the revision system. Zhang *et al.* (2019) investigated how the knowledge in answers becomes obsolete and identified the characteristics of such obsolete answers. Ford *et al.* (2018) proposed a mentorship program to Stack Overflow in which novice users get assistance with asking a question in an on-site help chat room. They found that the chat room substantially helps to improve the questions that were asked by the novice users.

Different from the prior studies which improve the question answering process by improving the quality of questions and answers, we study the impact of the bounty system on the question answering process in terms of the solving-likelihood and time for bounty questions. We provide users with insights on how to use bounties more effectively.

11 Conclusion

Stack Overflow introduced their bounty system in 2009 as a way of improving the solving-likelihood of questions. In this system, users can offer reputation points in exchange for an answer to their question.

In this paper, we studied 129,202 bounty questions (i.e., from Sep. 2011 to Aug. 2017) to study the impact of bounties on the solving-likelihood and solving-time of a question. In addition, we studied the most important factors for the solving-likelihood and solving-time of bounty questions. The main findings of our study are as follows:

1. Questions are likely to attract more traffic after receiving a bounty than non-bounty questions. In addition, bounty questions have a higher solving-likelihood than non-bounty questions, especially in very large communities with a relatively low question solving-likelihood.
2. Bounty questions with a higher bounty value have a higher solving-likelihood, however, a higher bounty value does not expedite the solving of a bounty question.
3. Long-standing unsolved questions with bounties are more likely to be solved than those without bounties. For example, the solving-likelihood of a question that has been unsolved for 100 days increases from 1.7% to 55% after proposing a bounty.

Our study shows that while bounties are not a silver bullet for getting a question solved, they are associated with a higher solving-likelihood of a question in most cases. In particular, when a question is asked in a community

(or tag) with a large number of active answerers, the chance of a bounty being successful is relatively high. As questions that are still unsolved after two days hardly receive any traffic, we recommend that Stack Overflow users propose a bounty as soon as possible after those two days for it to be the most successful. In addition, we see an opportunity for Stack Overflow to improve the bounty system by making recommendations to users who are about to propose a bounty about the tag(s) or bounty value that will give the question the highest solving-likelihood.

References

Ahasanuzzaman, M., Asaduzzaman, M., Roy, C. K., and Schneider, K. A. (2018). Classifying Stack Overflow posts on API issues. In *2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pages 244–254.

Anderson, A., Huttenlocher, D., Kleinberg, J., and Leskovec, J. (2012). Discovering value from community activity on focused question answering sites: A case study of Stack Overflow. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 850–858. ACM.

Anderson, A., Huttenlocher, D., Kleinberg, J., and Leskovec, J. (2013). Steering user behavior with badges. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, pages 95–106.

Bauer, D. F. (1972). Constructing confidence sets using rank statistics. *Journal of the American Statistical Association*, **67**(339), 687–690.

Bonferroni, C. (1936). Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commericiali di Firenze*, **8**, 3–62.

Cavusoglu, H., Li, Z., and Huang, K.-W. (2015). Can gamification motivate voluntary contributions?: The case of Stack Overflow Q&A community. In *Proceedings of the 18th ACM Conference Companion on Computer Supported Cooperative Work & Social Computing*, CSCW'15 Companion, pages 171–174.

Chen, C., Xing, Z., and Liu, Y. (2017). By the community & for the community: A deep learning approach to assist collaborative editing in Q&A sites. *Proc. ACM Hum.-Comput. Interact.*, **1**(CSCW), 32:1–32:21.

Chen, C., Chen, X., Sun, J., Xing, Z., and Li, G. (2018). Data-driven proactive policy assurance of post quality in community Q&A sites. *Proc. ACM Hum.-Comput. Interact.*, **2**(CSCW), 33:1–33:22.

Efron, B. (1986). How biased is the apparent error rate of a prediction rule? *Journal of the American Statistical Association*, **81**(394), 461–470.

Farrar, D. E. and Glauber, R. R. (1967). Multicollinearity in regression analysis: the problem revisited. *The Review of Economic and Statistics*, pages 92–107.

Finifter, M., Akhawe, D., and Wagner, D. (2013). An empirical study of vulnerability rewards programs. In *USENIX Security Symp.*, pages 273–288.

Ford, D., Lustig, K., Banks, J., and Parnin, C. (2018). "We don't do that here": How collaborative editing with mentors improves engagement in social Q&A communities. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 608:1–608:12.

Gwet, K. *et al.* (2002). Inter-rater reliability: dependency on trait prevalence and marginal homogeneity. *Statistical Methods for Inter-Rater Reliability Assessment Series*, **2**, 1–9.

Hanrahan, B. V., Convertino, G., and Nelson, L. (2012). Modeling problem difficulty and expertise in stackoverflow. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work Companion*, pages 91–94. ACM.

Harrell, Jr., F. E. (2006). *Regression modeling strategies*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Hata, H., Guo, M., and Babar, M. A. (2017). Understanding the heterogeneity of contributors in bug bounty programs. In *Proceedings of the 11th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, ESEM '17, pages 223–228.

Hsieh, G., Kraut, R. E., and Hudson, S. E. (2010). Why pay?: exploring how financial incentives are used for question & answer. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 305–314. ACM.

Jan, S. T., Wang, C., Zhang, Q., and Wang, G. (2017). Towards monetary incentives in social Q&A services. *arXiv preprint arXiv:1703.01333*.

Kanda, T., Guo, M., Hata, H., and Matsumoto, K. (2017). Towards understanding an open-source bounty: Analysis of bountysource. In *2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pages 577–578. IEEE.

Krishnamurthy, S. and Tripathi, A. K. (2006). Bounty programs in free/libre/open source software. In *The Economics of Open Source Software Development*, pages 165–183. Elsevier.

Liu, J., Zhou, P., Yang, Z., Liu, X., and Grundy, J. (2018). Fasttagrec: fast tag recommendation for software information sites. *Automated Software Engineering*, **25**(4), 675–701.

Long, J. D., Feng, D., and Cliff, N. (2003). Ordinal analysis of behavioral data. *Handbook of psychology*.

Maillart, T., Zhao, M., Grossklags, J., and Chuang, J. (2017). Given enough eyeballs, all bugs are shallow? Revisiting Eric Raymond with bug bounty programs. *Journal of Cybersecurity*, **3**(2), 81–90.

McIntosh, S., Kamei, Y., Adams, B., and Hassan, A. E. (2016). An empirical study of the impact of modern code review practices on software quality. *Empirical Software Engineering*, **21**(5), 2146–2189.

Munaiah, N. and Meneely, A. (2016). Vulnerability severity scoring and bounties: Why the disconnect? In *Proceedings of the 2nd International Workshop*

on Software Analytics, SWAN 2016, pages 8–14.

Nakasai, K., Hata, H., and Matsumoto, K. (2018). Are donation badges appealing? a case study of developer responses to eclipse bug reports. *IEEE Software*.

Ponzanelli, L., Mocci, A., Bacchelli, A., Lanza, M., and Fullerton, D. (2014). Improving low quality Stack Overflow post detection. In *2014 IEEE International Conference on Software Maintenance and Evolution*, pages 541–544.

Rajbahadur, G. K., Wang, S., Kamei, Y., and Hassan, A. E. (2017). The impact of using regression models to build defect classifiers. In *Proceedings of the 14th International Conference on Mining Software Repositories (MSR)*, pages 135–145.

Romano, J., Kromrey, J. D., Coraggio, J., and Skowronek, J. (2006). Appropriate statistics for ordinal level data: Should we really be using t-test and cohen'sd for evaluating group differences on the nsse and other surveys. In *annual meeting of the Florida Association of Institutional Research*, pages 1–33.

Srba, I. and Bielikova, M. (2016). Why is Stack Overflow failing? Preserving sustainability in community question answering. *IEEE Software*, **33**(4), 80–89.

Stack Exchange (2017). Stack Exchange. https://archive.org/details/stackexchange. (last visited: Dec. 20, 2017).

Stack Overflow (2019). Stack Overflow: User Privileges. https://stackoverflow.com/help/privileges. (last visited: Jan. 23, 2019).

Tantithamthavorn, C. and Hassan, A. E. (2018). An experience report on defect modelling in practice: Pitfalls and challenges. In *Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice*, pages 286–295. ACM.

Thongtanunam, P., McIntosh, S., Hassan, A. E., and Iida, H. (2016). Revisiting code ownership and its relationship with software quality in the scope of modern code review. In *Software Engineering (ICSE), 2016 IEEE/ACM 38th International Conference on*, pages 1039–1050. IEEE.

Tian, Y., Nagappan, M., Lo, D., and Hassan, A. E. (2015). What are the characteristics of high-rated apps? A case study on free android applications. In *Software Maintenance and Evolution (ICSME), 2015 IEEE International Conference on*, pages 301–310. IEEE.

Treude, C., Barzilay, O., and Storey, M.-A. (2011). How do programmers ask and answer questions on the web?: Nier track. In *Software Engineering (ICSE), 2011 33rd International Conference on*, pages 804–807. IEEE.

Wang, S., Lo, D., Vasilescu, B., and Serebrenik, A. (2014). Entagrec: An enhanced tag recommendation system for software information sites. In *2014 IEEE International Conference on Software Maintenance and Evolution*, pages 291–300. IEEE.

Wang, S., Lo, D., Vasilescu, B., and Serebrenik, A. (2018a). Entagrec ++: An enhanced tag recommendation system for software information sites. *Empirical Software Engineering*, **23**(2), 800–832.

Wang, S., Chen, T.-H. P., and Hassan, A. E. (2018b). How do users revise answers on technical Q&A websites? A case study on Stack Overflow. *IEEE Transactions on Software Engineering*.

Wang, S., Chen, T.-H., and Hassan, A. E. (2018c). Understanding the factors for fast answers in technical Q&A websites. *Empirical Software Engineering*, **23**(3), 1552–1593.

Wu, Y., Wang, S., Bezemer, C.-P., and Inoue, K. (2018). How do developers utilize source code from Stack Overflow? *Empirical Software Engineering*, pages 637–673.

Xia, X., Lo, D., Wang, X., and Zhou, B. (2013). Tag recommendation in software information sites. In *Proceedings of the 10th Working Conference on Mining Software Repositories, MSR '13, San Francisco, CA, USA, May 18-19, 2013*, pages 287–296.

Zhang, H., Wang, S., Chen, T.-H. P., and Hassan, A. E. (2019). An empirical study of obsolete answers on Stack Overflow. *IEEE Transactions on Software Engineering*.

Zhao, M., Grossklags, J., and Chen, K. (2014). An exploratory study of white hat behaviors in a web vulnerability disclosure program. In *Proc. of the Workshop on Security Information Workers*, pages 51–58. ACM.

Zhao, M., Laszka, A., and Grossklags, J. (2017). Devising effective policies for bug-bounty platforms and security vulnerability discovery. *Journal of Information Policy*, **7**, 372–418.

Zhou, J. (2019). Supplementary material for our paper. https://github.com/SAILResearch/wip-18-jiayuan-SO-bounty-SupportMaterials/blob/master/appendix.pdf.

Zhou, J., Wang, S., Bezemer, C.-P., Zou, Y., and Hassan, A. E. (2019). Bounties in open source development on github: A case study of bountysource bounties. *arXiv preprint arXiv:1904.02724*.