Asking questions is easy, asking great questions is hard: Constructing Effective Stack Overflow Questions

Jane Hsieh
jhsieh@oberlin.edu
Oberlin College
Oberlin, Ohio

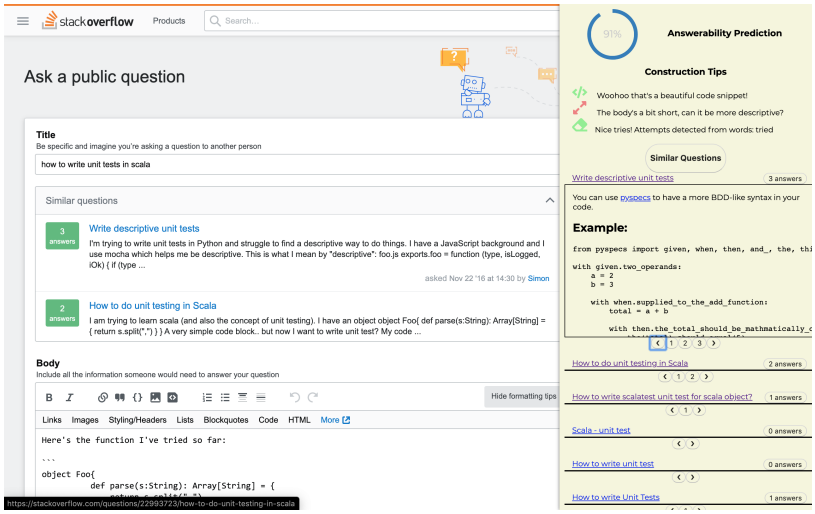

Figure 1: Dynamic Chrome Sidebar to help Improve Question Quality

ABSTRACT

This paper explores and seeks to improve the ways in which Stack Overflow question posts can elicit answers using existing data and studies. Using statistical data analysis approaches and reviews of existing literature, we pinpoint three key factors that are found in many previously successful / answerable questions. We then present a prototypical sidebar for the ask page that dynamically (1) evaluates the quality of questions in construction (2) displays answer previews of relevant questions (3) scaffolds the identified factors to subsequent askers during their question development processes.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

Social Q& A, Data Mining, Online Communities

ACM Reference Format:
Jane Hsieh. 2020. Asking questions is easy, asking great questions is hard: Constructing Effective Stack Overflow Questions . In *Proceedings of* . ACM, New York, NY, USA, 6 pages.

Unpublished working draft. Not for distribution.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00

1 INTRODUCTION

Stack Overflow has become one of the most well-known and fastest Q%A platform for programmers. However, it has also been identified as an environment that is hostile toward certain groups of users such as novices and women [6, 8]. While previous research has focused on making posts on such platforms comprehensible [2, 4, 7] or identifying the barriers that prevent users from contributing [6], few sources have consolidated positive qualities of good question posts into a format that is accessible and helpful toward potential question askers.

In this project, we aim to bridge this gap between awareness and implementation. To start off, we establish a metric for identifying quality Stack Overflow (SO) questions and utilize the large corpus of available questions to mine for qualities that are significantly correlated with good posts. To find potential candidates for these answer-eliciting factors, we review current literature to glean insight from findings of qualitative approaches and also mine from some of the more successful questions for trends and practices. We then help users incorporate these qualities into their question

formulation processes by injecting a sidebar to Stack Overflow's ask page in the form of a Chrome extension. The plugin

(1) scans the question body for presence of the identified criteria
(2) makes actionable suggestions to guide improvement
(3) evaluates the question's likelihood of receiving an answer in its current state
(4) embeds previews to answers of related questions to help construct context

2 MOTIVATION

2.1 Broad incentives

One of the factors incentivizing this project is the objective to help bridge the gender gap in online programming communities. As of 2016, it has been found that only 5.8% Stack Overflow users are women [6], and they make up for less than 5% of all open source contributions [5]. Furthermore, 5 of 14 of barriers ($\approx$ 35%) that have been found to impede Stack Overflow users are significantly problematic for females [6]. Therefore, the purpose of this study is to develop a tool with functionalities to help potential contributors and newcomers from a diverse set of backgrounds (especially those who are female) to overcome such barriers.

2.2 Targeting Question Quality

Most concretely, this project aims to achieve the larger scale goal by attacking the quality of composed questions (many studies in the past has focused on the quality of answer posts [2, 3, 7, 9, 10]). Recently, Stack Exchange has identified the need for more and better constructed questions on the Stack Overflow platform. Indeed, as of February 2020, the 30% of the questions on the site remains unanswered [1]. To mitigate the need for more (effectively constructed) questions, the SO platform has revised its reputation reward policy during November 2019: reputation awarded to each user for receiving an upvote to their question is now doubled from 5 to 10 points. Even users who received upvotes on their questions in the past is retroactively "refunded" these reputation points. Thus, it is imperative to not only identify the factors that help users construct more effective and answerable questions (in this paper, we define answerable questions as those with high likelihood of receiving an answer), but to also surface them to the users in a constructive and timely fashion.

3 RELATED WORKS

In this section we review bodies of literature that relate to the aforementioned research goals and motivates the design of certain features and interactions of the plugin. First we explore works analyzing the design and impact of the Stack Overflow. Next, we identify factors that affect the probability of a question post from receiving answers. To find possible candidates of such factors, we explore past qualitative studies to gather potentially relevant aspects of successful questions (in the data extraction section, we will employ a statistical technique to identify which of these factors are actually correlated with higher quality questions). Finally, we discuss the design implications resulting from these studies and how the current tool provides additional value to the current body of literature.

3.1 Current statistics

At the time of this writing (March 2020), SO has amassed around 12 million users, receives approximately 7 thousand questions submissions each day, and has a median answer time of 35 minutes[2]. But 30% of the questions on the site are also unanswered.

To help reduce this significant portion of unanswered questions, we seek to improve the quality of posted questions to raise their likelihood of receiving answers. One of the implications arising from the study examining barriers [6] is that the posting process can be enhanced by automatic feedback on the quality of the questions. Hence, we find ways of automatically detecting factors during the construction process to provide instantaneous feedback to potential question posters on SO. In the following section, we discuss the origins of the considered factors.

3.2 Roadblocks, Success Factors and Design Implications

To identify potential success factors for answerable questions as well as , we explored various features discussed in [1, 2, 6, 11]. One of the categories of barriers preventing new users from posting is the "On-ramp roadblocks" identified in [6]. These include factors such as difficulties with making sure that a duplicate post doesn't already exist, struggles with time constraints and self-confidence with expertise in the topic, and effort required to learn proper etiquette of the community. While the need for automatic feedback discussed in the previous section motivated the inclusion of a prediction about the answerabilty of a question, these barriers incentivized a section devoted to displaying of answers to related questions so that question writers can have access to existing posts to help them detect potentially duplicate questions, build context for their own question, and gain knowledge about community standards.

To identify qualities to avoid during question construction, we exploited factors from a study on *unanswered* questions on SO (which still occupies a significant portion of the site's corpus of questions, as was previously noted). Some of these questions can be too short in length and vague while others are too specific and without context. Others might be too hard to follow, difficult (technically), time-consuming, or simply a duplicate question. To compare this list of failure factors with successful characteristics, we also explored factors found in [2, 11], which includes the following:

- answerer's reputation / user identity
- presentation quality (signified by factors such as the presence of code snippets and linked urls)
- length of question content
- time and day of asking
- technology in question
- question type

To identify which of these are actually correlated with frequently answered and highly upvoted questions, we perform a two-proportions z-test on the sets of highly successful and unsuccessful questions.

[1] stackexchange.com/sites?view=list#users

[2] https://data.stackexchange.com/stackoverflow/query/9449

In the following section, we give a detailed account of our measurement of success as well as the way in which we queried for the datasets.

4 DATA EXTRACTION

Each post on SO can receive upvotes and downvotes. We leverage this user rating system to help measure the success / quality / popularity of question posts. The Stack Exchange Data Explorer allowed for querying large amounts of data from the site at a time, and had its own establishment of a score, which is simply upvotes - downvotes. In this study, our quantification of "success" utilizes the view count and votes to obtain a score for each question

$$\text{score}(Q) = \frac{\text{upvotes} - \text{downvotes}}{\text{view count}}$$

The following SQL query was composed in SEDE to find the sets of highly successful and unsuccessful questions. Note that in addition to taking the raw score, we also take the number of views on the question into account for popular questions (those that are inquiries to many) may be prone to receiving more votes of approval. These upvotes from the mass may reflect a common need for the question topic rather than effectiveness of the question composition. Finally, while we are aware of the “accepted" field of a question post, it is not utilized in this query for there are cases where answer posts do not receive the acceptance they deserve. This phenomenon may be due to various reasons on the part of the question’s author.

```
SELECT Top 30000 Questions.Id AS [Post Link],
       Questions.Id as Id,
       Questions.ViewCount as ViewCount,
       Questions.Score as Score,
       Questions.Body as Body,
       Questions.AcceptedAnswerId as AcceptedAnswerId,
       ROUND(1.0 * Questions.Score/Questions.ViewCount, 8) AS scTovc

FROM Posts AS Answers
    INNER JOIN Posts AS Questions
    ON Questions.Id = Answers.ParentId
    WHERE ROUND(1.0 * Questions.Score/Questions.ViewCount, 8) > 0.13

GROUP BY
Questions.Id,
Questions.ViewCount,
Questions.Score,
Questions.Body,
Questions.AcceptedAnswerId

ORDER BY NEWID();
```

The threshold for “successful posts" in this study are those with score greater than 0.13 and those needing improvement have score below −0.2. The choice of these specific thresholds did not serve a particular purpose other than obtaining data sets of approximately the same sizes (1057 and 1568).

5 VERIFICATION / FINDINGS

To extract the relevance of some factors listed in the literature review section, we used a two-proportions z-test to compare the samples. In the present study, we focused on the factors that were readily extractable from the text of the question body. These were *code snippet presence, length of question, presence of attempt-signifying words*, and the *presence of links*. In the case of length, we categorize an answer body to be of sufficient size (1) when it surpases the mean length of the successful questions, and too short (0) otherwise.

The two-proportions z-test was chosen for several reasons. First, each feature of our data can be captured by a binary value. Second, the data set is large enough so that we are able to choose a sampling size that allows the data set to be at least 20 times the size of each sample while ensuring that each sample contains at least 10 successes and 10 failures. Finally, we are able to perform simple random sampling and acquire independent samples, thus meeting all criteria required for a two-proportions z-test. The table below outlines how this method helps us determine whether the difference between two proportions is significant for the examined set of features.

	Code	Length	Attempt	Link
Confidence	99.92%	96.2%	82.2%	52.2%

These outcomes are a result of performing the z-test a set of 1057 successful questions and 1568 questions that needed improvements. The samples were randomly chosen sets of 52 questions each, and 1000 trials were performed. To pose the factors of attempt presence and length as binary variables, we choose the mean length (1350 characters) as the threshold for whether the post is lengthy (1) or not (0). And for questions that contain descriptions of previous attempts, we search through the post content for the following list of attempt-signifying words: $[attempt, try, tried, tries]$. From these results we have concluded that the three statistically significant factors are code presence, length and attempt-presence. In the following section we outline how these factors are used to predict the answerability of a question that is under construction.

5.1 Classification

To perform binary classification on SO questions, we used the identified three factors as features to train a binary logistic regression model since the target variable (answeredness) is dichotomous. To train, we used smaller set of questions, consisting of the top 350 posts with the highest scores and 486 questions with the lowest scores. The two binary classes are (1) *answered* and (0) *not answered*. The output of the model produces coefficients for the logit function

$$p(\vec{X}) = \frac{1}{1 + e^{-(\beta_0 0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3)}}$$

is then used to make a prediction about the probability of a posed question being answered based on the presence or absence of the three examined factors. In our case the first term of the coefficient vector is left out since it ends up a constant, so effectively we have

$$\vec{\beta} = (\beta_1, \beta_2, \beta_3) \approx (2.144, 0.126, 0.223)$$

These coefficient are then (matrix) multiplied by the corresponding feature variables, signified by the vector

$$\vec{X} = (X_1, X_2, X_3)$$

where

$$X_1 = \text{code snippet presence}$$

$$X_2 = \text{median length surpassed}$$

$$X_3 = \text{presence of attempt-signifying words}$$

In the training process, we divided our data into test (25%) and training (75%) sets to evaluate performance. We evaluate the performance of the model using a confusion matrix, which is visualized below as a heatmap. The diagonals of the heatmap represent correct hits from the model (true negatives and positives) whereas the top

right represents the number of false positives and the bottom left displays the amount of false negatives.

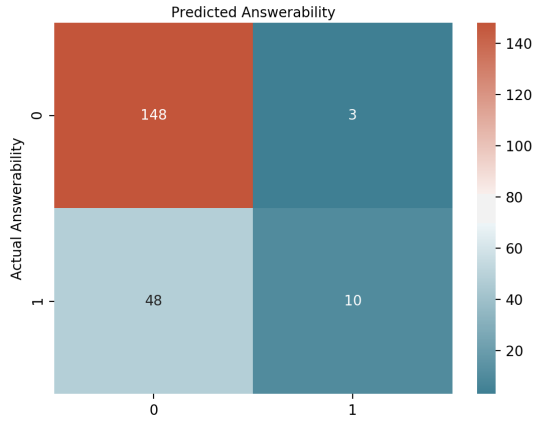

Figure 2: Heatmap of Model's Confusion Matrix

From the confusion matrix we can also retrieve the following performance measures for the model:

Precision	Accuracy	Recall
76.92%	75.598%	17.24%

Even though the recall rate is lower than one can hope for, the classification rate / accuracy (the proportion of cases that were correctly classified) and precision (proportion of positives that are true) of the model are both acceptable.

Finally, the output probability of the logit function is used to give our estimated likelihood that the post of concern will receive an answer, given its feature vector.

6 TOOL FEATURES AND IMPLEMENTATION

The ask page of SO currently includes a section containing general tips and suggestions for how to "Draft your question". However, it is not targeted to a particular post or even type of question can cite here many papers that examine different categories of questions, including my own publication!!, and fails to offer input based on the progress of the user. Furthermore, while users are prevented from submitting questions if certain fields are missing, they are not notified of missing pieces to their composed inquiry prior to hitting the submit button. This type of negative feedback without warning may discourage first-time askers from composing questions in the future, or even prevent them from complete their current posts!

To help alleviate some of these issues of the design page, our tool focuses on providing the following value-adds:

(1) Display a course-grained but digestible estimation for the likelihood that the question post under construction will receive an answer, based on the features that we've found to be significant factors
(2) Give specific suggestions for improving the user's current content in a targeted and encouraging way
(3) Show answers of similar questions to help the user
  (a) build context for their own inquiry
  (b) learn etiquette of the community
  (c) avoid duplicating an existing question

The plugin is built using a React Chrome Sidebar boilerplate and the project repo can be found on github.com/janeon/honors-plugin. The extension is intended to be used only on the ask page of the Stack Overflow site, and below we describe its features and outline the key components of its implementation.

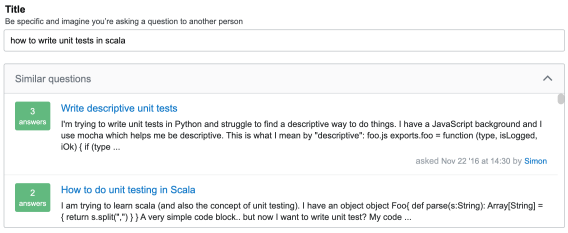

Figure 3: Current Features: Similar Questions

6.1 Current State of SO's Ask Page

As stated previously, the ask page does not update their suggestions based on user inputs, but it does contain a scrollable "Similar questions" section that is dynamically populated based on the inputted title. However this feature is incomplete - much content about each question is lost, perhaps due to limitations posed by available real estate and an effort to keep each question debrief concise. These missing pieces include the full body of the question and the answer posts. In section 6.5 we outline the how the absence of these factors are mitigated by the *Answer Previews* section of our tool.

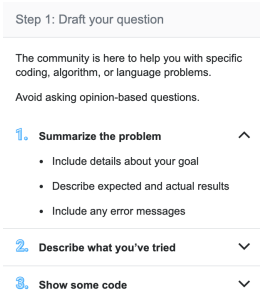

Figure 4: Current Features: Suggestions

6.2 Persuasive Design

THis is the persuasive design section
THis is the persuasive design section
THis is the persuasive design section
THis is the persuasive design section
THis is the persuasive design section

6.3 Answerability Prediction

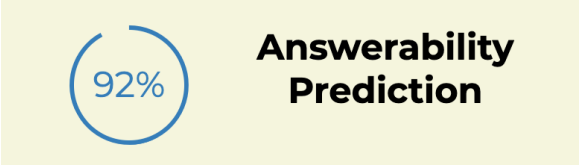

Figure 5: The Dynamic Answerability Prediction Gamifies the Process of Crafting an Effective Question

Using the results from the findings section (code snippet presence, length, and description of past attempts are all factors that correlate with a quality and answerable question) and the logistic regression model from section 5.1, we are able to extract an answerability probability to help users gauge the current effectiveness of their question.

To help further visualize the estimated answerability, we display the probability as a percentage inside a radial progress bar. Since this progress bar and the implicated estimation updates dynamically based on changes in the body content, this feature is included to give users an additional sense of agency toward actively improving the quality of their question post.

6.4 Improvement Recommendations

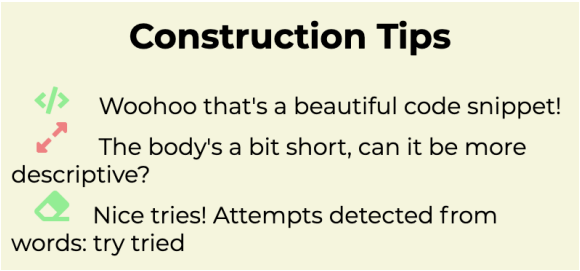

Figure 6: Realtime Suggestions Allow Users to Improve Question Content on the Spot

In addition to providing an estimated likelihood of a question's current answerability, we also give specific suggestions after detecting the presence or absence of the identified factors. When the user has already successfully included a feature, the embedded text is a congratulatory comment, and the associated icon is colored green. However, if the feature is found to be absent, its icon takes on an alarming red and a suggestive note is made to encourage inclusion of the feature.

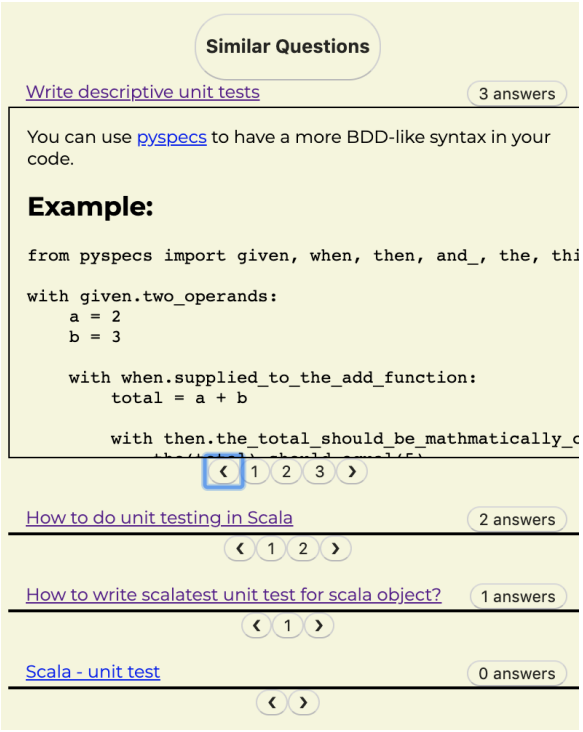

Figure 7: Answers of Similar Questions

6.5 Answer Previews

An additional feature we offer as a part of this tool is the ability to quickly debrief answers to similar questions. Utilizing the existing similar questions section to our advantage, we parse the html element of each question to obtain its question ID via its corresponding hyperlink. Using the question ID, we query the Stack Exchange API using axios requests to acquire the answer IDs associated with the question's answer posts.

Each answer is then displayed in a slideshow format in this last section to allow easy browsing through various answers without compromising too much the amount of content displayed for an answer at a time (each box for displaying answers is also scrollable in both directions to accomodate overflowing content). To avoid information overload and getting lost in the answer posts of many different questions, each question's answer section is made collapsible, so that the user has the flexibility of focusing on a single or multiple related questions at a time.

The entirety of the section is also foldable so that users who feel overwhelmed by the large (but organized) amounts of information have the option of reducing the presented content. Finally, each question embeds the link to the page of the related question in case the user needs access to the page in full. Some additional information and features that users may seek out include comments to the question and answer posts, accepted answer status, upvotes and downvotes, as well as any bounty awarded to certain answers.

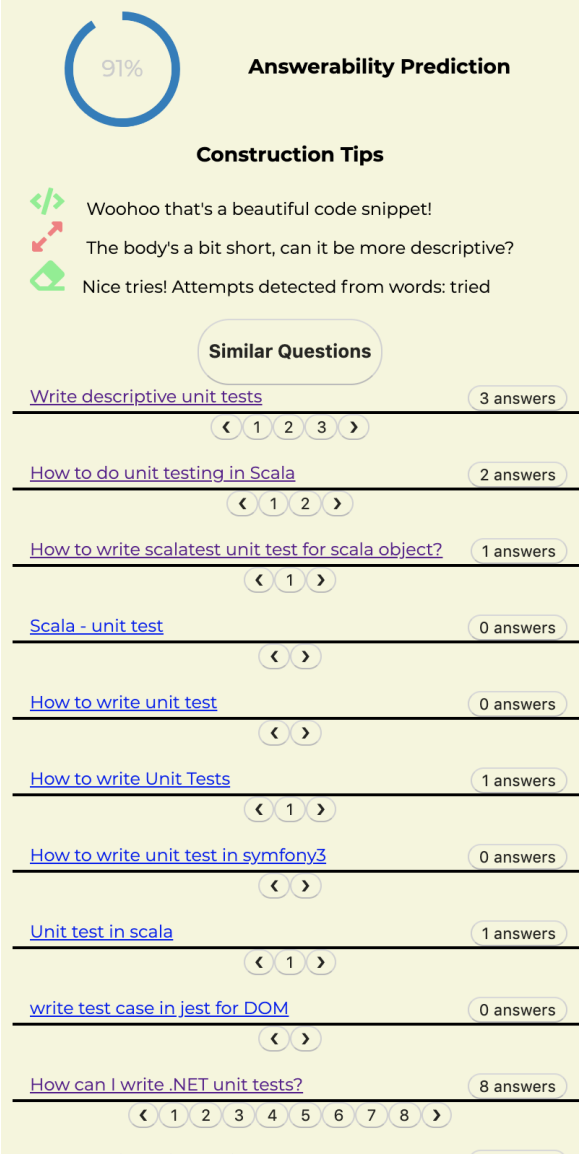

Figure 8: Collapsible Answers to Save on Real Estate

7 CONCLUSIONS AND FUTURE WORK

In this study, we have employed a two-proportions z-test to extract some properties common to successful or answerable questions, and trained a logistic regression to recognize a successfully constructed question using these found criteria. We then applied these results from the statistical methods toward building a Chrome extension to help scaffold some of these practices to new users of the site (particularly the ask page).

In designing the tool, we could have completed more field work examining feedback from actual users to engage and pinpoint some of the real-life painpoints. These were not achieved due to the limitations of time accessibility to such users. Similarly, user testing (both retrospective and during the time of development) would have provided valuable feedback toward improving its applicability and usability, but such studies were not conducted for similar reasons, and is to be prioritized as future goals.

Finally, we hope to examine in closer detail how these and other factors might affect (sustained) participation by populations who are traditionally less involved in computing. It was an initial goal to address several of the barriers discussed in section 3.2, and further testing and improvements are required before determining the effectiveness of these features in combating such roadblocks.

8 THREATS TO VALIDITY

(1) data sampling
(2) lack of cross validation

ACKNOWLEDGMENTS

REFERENCES

[1] Muhammad Asaduzzaman, Ahmed Shah Mashiyat, Chanchal K Roy, and Kevin A Schneider. 2013. Answering questions about unanswered questions of stack overflow. In *2013 10th Working Conference on Mining Software Repositories (MSR)*. IEEE, 97–100.

[2] F. Calefato, F. Lanubile, M. C. Marasciulo, and N. Novielli. 2015. Mining Successful Answers in Stack Overflow. In *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories*. 430–433. https://doi.org/10.1109/MSR.2015.56

[3] Fabio Calefato, Filippo Lanubile, and Nicole Novielli. 2016. Moving to stack overflow: Best-answer prediction in legacy developer forums. In *Proceedings of the 10th ACM/IEEE international symposium on empirical software engineering and measurement*. 1–10.

[4] D Chang, Michael Schiff, and Wei Wu. 2013. Eliciting Answers on StackOverflow. *Working Paper* (2013).

[5] Paul A David and Joseph S Shapiro. 2008. Community-based production of open-source software: What do we know about the developers who participate? *Information Economics and Policy* 20, 4 (2008), 364–398.

[6] Denae Ford, Justin Smith, Philip J. Guo, and Chris Parnin. 2016. Paradise Unplugged: Identifying Barriers for Female Participation on Stack Overflow. In *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering* (Seattle, WA, USA) *(FSE 2016)*. Association for Computing Machinery, New York, NY, USA, 846–857. https://doi.org/10.1145/2950290.2950331

[7] Kerry Hart and Anita Sarma. 2014. Perceptions of Answer Quality in an Online Technical Question and Answer Forum. In *Proceedings of the 7th International Workshop on Cooperative and Human Aspects of Software Engineering* (Hyderabad, India) *(CHASE 2014)*. Association for Computing Machinery, New York, NY, USA, 103–106. https://doi.org/10.1145/2593702.2593703

[8] Lena Mamykina, Bella Manoim, Manas Mittal, George Hripcsak, and Björn Hartmann. 2011. Design Lessons from the Fastest Q&a Site in the West. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI '11)*. Association for Computing Machinery, New York, NY, USA, 2857–2866. https://doi.org/10.1145/1978942.1979366

[9] Qiongjie Tian, Peng Zhang, and Baoxin Li. 2013. Towards predicting the best answers in community-based question-answering services. In *Seventh International AAAI Conference on Weblogs and Social Media*.

[10] Yuan Tian, Pavneet Singh Kochhar, Ee-Peng Lim, Feida Zhu, and David Lo. 2013. Predicting best answerers for new questions: An approach leveraging topic modeling and collaborative voting. In *International Conference on Social Informatics*. Springer, 55–68.

[11] Christoph Treude, Ohad Barzilay, and Margaret-Anne Storey. 2011. How do programmers ask and answer questions on the web?(NIER track). In *Proceedings of the 33rd international conference on software engineering*. 804–807.