Early Detection and Guidelines to Improve Unanswered Questions on Stack Overflow

Saikat Mondal
University of Saskatchewan, Canada
saikat.mondal@usask.ca

C M Khaled Saifullah
University of Saskatchewan, Canada
khaled.saifullah@usask.ca

Avijit Bhattacharjee
University of Saskatchewan, Canada
avijit.bhattacharjee@usask.ca

Mohammad Masudur Rahman
Dalhousie University, Canada
masud.rahman@dal.ca

Chanchal K. Roy
University of Saskatchewan, Canada
chanchal.roy@usask.ca

ABSTRACT

Stack Overflow is one of the largest and most popular question-answering (Q&A) websites. It accumulates millions of programming related questions and answers to support the developers in software development. Unfortunately, a large number of questions are not answered at all, which might hurt the quality or purpose of this community-oriented knowledge base. Up to 29% of Stack Overflow questions do not have any answers. There have been existing attempts in detecting the unanswered questions. Unfortunately, they primarily rely on the question attributes (e.g., score, view count) that are not available during the submission of a question. Detection of the potentially unanswered questions in advance during question submission could help one improve the question and thus receive the answers in time. In this paper, we compare unanswered and answered questions quantitatively and qualitatively by analyzing a total of 4.8 million questions from Stack Overflow. We find that topics discussed in the question, the experience of the question submitter, and readability of question texts could often determine whether a question would be answered or not. Our qualitative study also reveals several other non-trivial factors that not only explain (partially) why the questions remain unanswered but also guide the novice users to improve their questions. We develop four machine learning models to predict the unanswered questions during their submission. According to the experiments, our models predict the unanswered questions with a maximum of about 79% accuracy and significantly outperform the state-of-the-art prediction models.

CCS CONCEPTS

• **Software and its engineering → Software repository mining**; **Software maintenance and evolution**; **Software defect analysis**; **Software maintenance tools**;

KEYWORDS

Stack Overflow, unanswered questions, machine learning, prediction model, question attributes

ISEC 2021, February 25-27, 2021, Bhubaneswar, India

ACM ISBN XXX-X-XXXX-XXXX-X/XX/XX…$15.00
https://doi.org/XX.XXXX/XXXXXXX.XXXXXXX

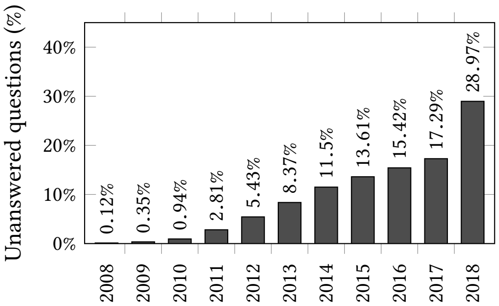


Figure 1: Percentage of the questions not answered in Stack Overflow over eleven years [23].

ACM Reference format:
Saikat Mondal, C M Khaled Saifullah, Avijit Bhattacharjee, Mohammad Masudur Rahman, and Chanchal K. Roy. 2021. Early Detection and Guidelines to Improve Unanswered Questions on Stack Overflow. In *Proceedings of ISEC '21: Innovations in Software Engineering Conference, Bhubaneswar, India, February 25-27, 2021 (ISEC 2021),* 11 pages.
https://doi.org/XX.XXXX/XXXXXXX.XXXXXXX

1 INTRODUCTION

Software developers often search for solutions on the web for their programming problems. The advent of Q&A websites such as Stack Overflow (henceforth, SO) has shaped the way developers search for information on the web [26]. SO is the largest and most popular Q&A site that accumulates millions of programming related questions and answers. These questions and answers are consulted by a large technical community of more than eight million users (as of December 2018 [23]). However, despite having such a large and engaged community, about 29% of SO questions remain unanswered [23]. Besides, this percentage has been gradually increasing over the years (e.g., Fig. 1). Unanswered questions devalue a knowledge base in terms of quality and relevance. The lack of answers to the programming related questions also impedes the normal development progress. *Early detection* of a question that might not be answered helps one improve the question and thus reduces the answering time. Early detection means detection of the potentially unanswered questions in advance during a question submission.

Several existing studies [3, 9, 29] investigate the unanswered questions of SO. Asaduzzaman et al. [3] manually analyze 400

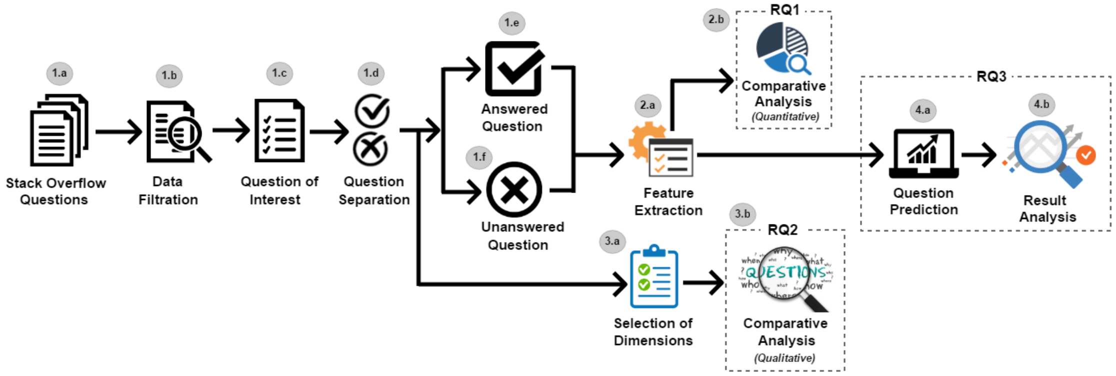


Figure 2: Schematic diagram of our conducted study.

questions of SO and report 13 factors (e.g., proprietary technology, irregular users) that might explain why the questions remain unanswered. However, many of their reported characteristics (e.g., impatient users) are hard to measure in practice. Besides, the authors also do not examine whether their features could predict the unanswered questions or not. Saha et al. [29] introduce machine learning models to classify answered and unanswered questions. Unfortunately, majority of their strong features (e.g., view count, score) are not available during the submission of a question. Thus, their model might not be able to predict the unanswered questions reliably during their submission. Calefato et al. [9] study such factors that might increase the chance of getting accepted answers to the SO questions. They suggest that high quality presentation (e.g., body length), positive sentiment, question submission in weekdays, and high user reputation might improve the chance of getting answers. However, when they were analyzed (in section 3.3), none of these factors (except user reputation) was found powerful enough to reliably predict the unanswered questions during their submission. In short, prediction of the unanswered questions is an open research problem that warrants further investigation.

In this paper, we (1) present a comparative analysis between answered and unanswered questions of SO, and (2) provide four machine learning models to predict the unanswered questions. We conduct our study in three major phases. First, we analyze eight attributes of 4.8 million questions and their corresponding question submitters, and perform comparative analysis to determine how the unanswered questions differ from the answered questions. We find that the topics discussed in the question, experience of the question submitter, and readability of question texts could potentially determine whether a question would receive answers or not. Second, we manually investigate 400 questions (200 unanswered + 200 answered), and revisit the quantitative findings above to see whether the quantitative and qualitative findings support each other or not. Our qualitative findings not only confirm the quantitative findings above but also reveal several new insights. Third, we develop four machine learning models using the features of the questions that are available during their submission, and predict the unanswered questions. We choose four non-linear machine learning algorithms – Decision Trees (CART) [7], Random Forest (RF) [6], K-Nearest Neighbors (KNN) [1], and Artificial Neural Network (ANN) [33] – to train our prediction models. According to extensive experiments, our models predict the unanswered questions with a maximum accuracy of 79% which is highly promising. Comparison with two state-of-the-art models [9, 29] suggests that our models outperform them with statistically significant margins. We thus answer three research questions with our study as follows.

RQ1. How do the unanswered questions differ from the answered questions in terms of their texts and their submitters' activities? We conduct a comparative analysis between the unanswered and answered questions using eight quantitative metrics from the literature. We find that the topics of a question, the experience of a question submitter, and the readability of question texts could potentially determine whether a question would be answered or not. According to our analysis, questions related to programming IDEs (e.g., RStudio), libraries (e.g., TensorFlow), and frameworks (e.g., Angular6) often remain unanswered. Such questions are often discouraged at Stack Overflow since they attract subjective opinions rather than authentic answers. Second, the users who have a higher reputation score are more likely to get answers to their questions. Third, easy-to-read questions are more likely to be answered.

RQ2. How do the unanswered and answered questions differ qualitatively? Do the qualitative findings agree with the quantitative findings? We conduct a manual investigation of 400 questions (200 unanswered + 200 answered) to see whether there is an agreement between the qualitative and quantitative findings. Besides supporting all the quantitative findings, our qualitative study also reveals several other non-trivial factors that partially explain why the questions remain unanswered. For example, questions that discuss outdated technology, multiple technical issues, or do not include appropriate code examples in their texts are more likely to be not answered. Our findings also align with the guidelines posted by SO about asking a good question [21]. However, their guidelines are theoretical. On the contrary, our findings and insights are backed up by empirical evidence that can explain why certain questions are likely to receive answers and the others are not. Moreover, our models can help the developers improve their questions with instant predictions on their quality. Several questions receive working solutions in their comment thread. Thus, although they seem to be unanswered, they are actually answered.

RQ3. Can we predict the unanswered questions of Stack Overflow during their submission? We develop four machine learning models to predict the unanswered questions during their submission. Our models predict the unanswered questions with a maximum of about 79% accuracy which is significantly higher than that of the state-of-the-art models.

2 STUDY METHODOLOGY

In Fig. 2, we describe our overall methodology to answer the three research questions. We describe the steps below.

Step 1: Data Collection and Preprocessing. Table 1 shows the summary of our collected dataset. We collect SO questions using StackExchange Data API [23] (e.g., Fig. 2, Step 1.a). In particular, we collect questions related to four popular programming languages - C#, Java, JavaScript, and Python. We choose the questions that were submitted on the site in the year 2017 or earlier (e.g., Fig. 2, Step 1.b). We wanted to ensure enough time for each question to be assessed by the SO community. We get a total of 4,814,900 questions where 4,324,252 questions have one or more answers, and the remaining 500,648 questions are unanswered (e.g., Fig. 2, Steps 1.c − 1.f). To calculate user reputation, we collect the information of the SO users and the votes they received on their questions and answers. We also collect the data associated with SO tags assigned to the questions.

Table 1: Summary of the study dataset

	C#	Java	JavaScript	Python	Total
Answered	1,034,516	1,182,921	1,340,016	766,799	4,324,252
Unanswered	113,903	143,757	158,344	84,644	500,648
Total	1,148,419	1,326,678	1,498,360	851,443	**4,814,900**

Step 2: Compare Unanswered and Answered Questions. We extract eight features (e.g., Text Readability, Topic Response Ratio, Topic Entropy) from both unanswered and answered questions (e.g., Fig. 2, Step 2.a), and conduct a comparative analysis (quantitative) between the attributes of these two categories (e.g., Fig. 2, Step 2.b).

Step 3: Agreement Analysis. We manually analyze 400 questions (200 unanswered + 200 answered) to investigate the qualitative difference between unanswered and answered questions (e.g., Fig. 2, Steps 3.a, 3.b). In particular, we attempt to determine whether the qualitative findings agree with the quantitative findings or not.

Step 4: Predict Unanswered Questions. We build four machine learning models to predict whether the SO questions might be answered or not (e.g., Fig. 2, Step 4.a). Finally, we analyze the results of our models and also compare with the state-of-the-art models (e.g., Fig. 2, Step 4.b).

3 STUDY FINDINGS

We ask three research questions in this study. In this section, we answer them carefully with the help of our empirical and qualitative findings as follows:

3.1 Answering RQ1 : Comparison between Unanswered and Answered Questions using Quantitative Features

In this section, we contrast between the unanswered and answered questions using eight metrics that capture different aspects (e.g., readability, topic) of a question. Each of them is discussed as follows.

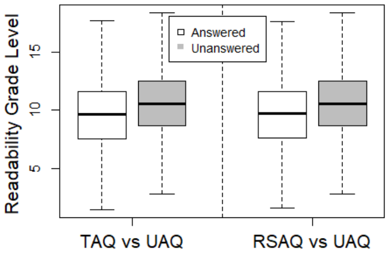


Figure 3: Text readability (TAQ=Total Answered Questions, RSAQ=Randomly Sampled Answered Questions, UAQ=Unanswered Questions).

(M1) Text Readability. Readability of a text document depends on the complexity of its vocabulary, syntax and the presentation style (e.g., line lengths, word lengths, sentence lengths, syllable counts) [34]. We extract the title and textual description from the body of each question to measure its text readability. The existing readability metrics are not well designed for handling the code or program elements within the texts [25]. We thus discard the code elements, code segments, and stack traces from the textual description using specialized HTML tags. For example, code elements are enclosed by *<code>* tag, code segments and stack traces are enclosed by *<code>* under *<pre>* tag. Unfortunately, some users do not use appropriate tags. We thus attempt to remove the code elements (e.g., method name) and stack traces from texts by employing appropriate regular expressions, as used by the existing literature [18]. We compute five popular readability metrics: *Automated Reading Index (ARI) [30], Coleman Liau Index [11], SMOG Grade [16], Gunning Fox Index [14], Readability Index (RIX) [2].* These metrics are widely used to estimate the comprehension difficulty of the English language texts [26]. First, we calculate the individual score (i.e., grade level) for each of the five metrics, and then determine the average readability of each question's text. Here, the lower the grade level, the easier the text is to read, and conversely, the higher the grade level, the more difficult the text is to read.

As shown in Fig. 3, we find a significant difference in the text readability between answered and unanswered questions. Answered questions have higher text readability (i.e., lower grade level) than that of the unanswered questions. Unanswered questions often contain long, multi-syllable words whereas the answered questions generally contain small, simple words. For example, according to our analysis, the average number of long (i.e., $length > 6$) and multi-syllable words per answered question are 17.8 and 27.8, whereas the same numbers for the unanswered question are 20.8 and 32.4 respectively. Since word length and syllable are important dimensions of the traditional readability tools [2, 11, 14, 16, 30], the low readability of the unanswered questions is pretty much explained. We also address the data imbalance issue during our comparative analysis. Since the number of answered questions is higher than that of the unanswered questions in our dataset, we randomly under-sample the answered questions to balance the dataset. Then we compare the readability scores between unanswered and answered questions again with the sampled data. Interestingly, we were able to reproduce the same finding as above. We also perform statistical significance tests such as Mann-Whitney-Wilcoxon and Cliff's Delta, and found statistical significance for all four programming languages (i.e., p-value = 0 <0.01 and Cliff's |d| = 0.17 (*small*)). That is, the readability of the answered questions is significantly higher than that of the unanswered questions regardless of the programming languages. Since we perform multiple comparisons (e.g., comparisons of unanswered, total answered and random sampling answered) our result could be affected by the type I error in null hypothesis testing. To mitigate this problem, we control the false discovery rate (FDR) by adjusting the p-values based on the method of Benjamini and Yekutieli [4].

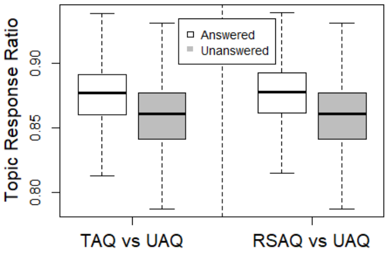


Figure 4: Topic response ratio (TAQ=Total Answered Questions, RSAQ=Randomly Sampled Answered Questions, UAQ=Unanswered Questions).

(M2) Topic Response Ratio. In SO, questions often might not receive answers due to their topical difficulty. Questions related to certain topics (e.g., software libraries, platform, IDE) usually get less attention than others (e.g., jquery-selectors, lisp) in SO. Questions on these topics might attract subjective opinions rather than the authentic answers. Thus, these questions are discouraged and often redirected to other community websites (e.g., GitHub) for appropriate answers [19].

We first examine which topics are more likely to receive answers and which are not. In SO, tags often capture the topics of a submitted question. Each question can have at most five tags and must have at least one tag [35]. We first calculate the answering ratio of each topic, R_t as follows.

$$R_t = \frac{A_Q}{T_Q} \tag{1}$$

where A_Q and T_Q represent the number of answered questions and total questions associated with the topic respectively. Since each question can have multiple topics (tags), we then measure the topic response ratio (Q_{resp}) of each question as follows.

$$Q_{resp} = \frac{1}{N} \times \sum_{i=1}^{N} R_{t_i} \tag{2}$$

where N is the number of topics associated with a question and R_{t_i} denotes the response ratio of the i^{th} topic of the question. According to our investigation, as shown in Fig. 4, the topic response ratio of the unanswered questions is lower than that of the answered questions. It indicates that the topics of the unanswered questions are of little interest to the SO user community. It is also possible that there are not sufficient experts to answer the topics discussed in the unanswered questions. We also found this difference as statistically significant using Mann-Whitney-Wilcoxon and Cliff's delta statistical test (i.e., p-value = 0 <0.01 and Cliff's d = 0.40 (*medium*)).

(M3) Topic Entropy. Single tags (e.g., java) assigned to a question often fail to explain the topics or subject matter of the question. On the contrary, multiple tags attached to a question could help the users better understand the topics and thus help them find the questions of their interest. We analyze the ambiguity of question topics and attempt to determine whether the unanswered questions use more ambiguous topics than those of the answered questions. In Information Theory, entropy is used as a measure of uncertainty of a random variable that takes up multiple values [28]. Similarly, we consider the probability of a question discussing a certain topic as a random variable, and determine topic entropy of each question. To measure the topic entropy of each question, we first calculate the prior probability (P_k) of each topic (i.e., tag) associated with the question at SO. P_k is defined as a ratio between the number of questions discussing a topic $k \in n$ and the number of questions discussing all the topics of SO. We then determine the topic entropy for each question as follows.

$$TE = -\frac{1}{n}\left[\sum_{k=1}^{n} P_k \times log(P_k)\right] \tag{3}$$

where n denotes the total number of topics of a question. Here, we calculate the average entropy (dividing the entropy by n) since the number of tags differ from question to question. If a question of SO uses one or more ambiguous topics, the entropy becomes higher.

We contrast between unanswered and answered questions in terms of their topic entropy. Table 2 summarizes the comparative analysis. We see that topic entropy of answered questions is significantly lower than that of the unanswered questions. Such a result indicates that the unanswered questions are more ambiguous than the answered questions, which possibly left them with no answers.

(M4) Metric Entropy and (M5) Average Terms Entropy. We use Metric Entropy and Average Terms Entropy to estimate the randomness of terms used in the textual part of the unanswered and answered questions. Metric Entropy is the Shannon entropy [12] divided by the character length of the question's text. It represents the randomness of the terms against a question [26]. Average Terms Entropy also estimates the randomness of each term for the question's text. However, in this case, the entropy of each term is calculated against all the questions on SO. We first calculate the entropy for each term in the SO data dump of December 2017. We then determine the equivalent entropy of each term for a question's text against all the questions. Finally, we calculate the Average Term Entropy for each question by summing each of the terms entropy for the question divide by the character length of the question's text. The entropy value describes the discriminating power of a term [26]. The lower the entropy of a question, the higher the use of uncommon terms in the question.

As shown in Table 2, we find a relatively lower metric and average terms entropy for the unanswered questions than that of the

Table 2: Comparison between unanswered and answered questions using quantitative features

Feature	Answered Question			Unanswered Question			MWW Test (p-value)	Cliff's |d|
	Mean	Median	Std. Dev.	Mean	Median	Std. Dev.		
Text Readability	9.89	9.67	4.12	11.05	10.53	4.90	0	0.17 (*small*)
Topic Response Ratio	0.88	0.88	0.04	0.86	0.86	0.05	0	0.40 (*medium*)
Topic Entropy	2.63	2.66	0. 67	2.73	2.75	0.67	0	0.09 (*negligible*)
Metric Entropy	0.0068	0.0064	0.0026	0.0061	0.0058	0.0023	0	0.17 (*small*)
Average Terms Entropy	0.0004	0.0004	0.0001	0.0004	0.0003	0.0001	0	0.09 (*negligible*)
Text-code Ratio	1.35	0.46	4.00	1.17	0.36	3.85	0	0.09 (*negligible*)
User Reputation	62.97	22.0	131.73	39.16	10.0	110.76	0	0.24 (*small*)
Received Response Ratio	0.71	1.0	0.45	0.60	0.87	0.45	0	0.30 (*small*)

answered questions. That is, the unanswered questions use more uncommon terms in their texts than the answered questions do. It indicates that the unanswered questions might use inappropriate terms that do not describe the problem correctly. Moreover, they might touch such topics that often remain unanswered. According to our analysis, we found the differences are statistically significant (i.e., p = 0 <0.01) using the Mann-Whitney-Wilcoxon test. However, the effect size is either small or negligible.

(M6) Text-code Ratio. The ratio between text and code of a question is often considered as an important dimension of question quality [31]. Large code segments with little explanation might make a question hard to comprehend. We thus wanted to find out whether the unanswered and answered question differ from each other in terms of their text-code ratio. We extract the code segments and texts from the body of a question. Then we calculate the text-code ratio by dividing the character length of the text with the length of the code segment. When the code segment is absent from a question, we assign a large number as the text-code ratio.

According to our analysis, the text-code ratio of the unanswered questions is lower than that of the answered questions (as shown in Table 2). That is, insufficient explanation of the code segment could be a potential factor that might explain why the questions remain unanswered. We also measure whether the difference of text-code ratio between unanswered and answered questions is statistically significant. From the Mann-Whitney-Wilcoxon test, we find significant p-value (p-value = 0 <0.01). However, the effect size from the Cliff's Delta test was found negligible.

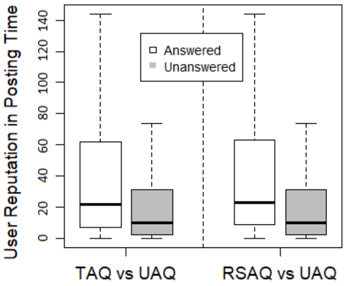


Figure 5: User reputation during question submission time (TAQ=Total Answered Questions, RSAQ=Randomly Sampled Answered Questions, UAQ=Unanswered Questions).

(M7) User Reputation. The reputation system of SO is designed to incentivize contributions and to allow assessment from the users [8]. Asaduzzaman et al. [3] argue that a question submitted by a user with a higher reputation has a higher possibility of getting answers. We thus consider the users' reputation as a distinctive feature of SO questions. The official data dump only reports the latest reputation scores of the users, which might not be appropriate for our analysis. However, SO stores all the activities (e.g., votes, acceptances, bounties) of users with their dates. We thus use the snapshot of the activities of users to calculate the reputation during their question submission. For reputation calculation, we used a standard equation provided by the SO [13].

According to our analysis (Fig. 5), the reputation of the authors of unanswered questions is lower than that of the authors of answered questions. It confirms that the high reputation score increases the chance of getting answers. Using the Mann-Whitney-Wilcoxon test, we find a significant p-value (i.e., p-value = 0 <0.01) although the effect size is small (i.e., Cliff's d = 0.24).

(M8) Received Response Ratio. The past question-answering history of a user might provide useful information as to whether a question submitted by the user will be answered or not [29]. Thus we investigate a user's question-answering statistics. In particular, we measure the percentage of answered questions (AQ_p) asked by users during their new question submission as follows.

$$AQ_p = \frac{\sum_{t<T} A_q}{\sum_{t<T} T_q} \times 100 \tag{4}$$

where T is the submission time of the question currently being processed, t denotes the submission time of the previous questions, A_q and T_q represent the total number of answered questions and asked questions respectively.

According to our analysis, a user with a higher percentage of receiving answers in the past has a higher chance of receiving answers in the future. As shown in Table 2, the authors of the answered questions received answers for about 71% of their submitted questions in the past. On the contrary, the same percentage is about 60% for the authors of the unanswered questions. The difference is statically significant (i.e., p-value = 0 <0.01) with a small effect size (i.e., Cliff's d = 0.30).

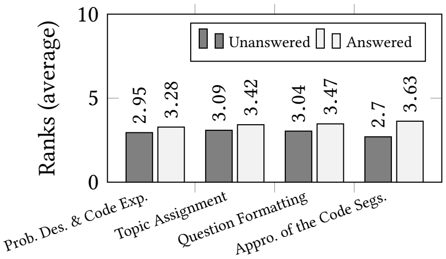


Figure 6: Comparison (qualitative) between unanswered and answered questions.

3.2 Answering RQ2 : Agreement between Quantitative and Qualitative Findings

While our quantitative analysis provides enough evidence that unanswered questions are significantly different from the answered questions, we further conduct qualitative analysis to better understand their difference. In particular, we attempt to determine whether the qualitative findings agree with the quantitative findings from the above section (Sec. 3.1) or not.

Study Setup. In SO, the topics of a question are often conveyed through predefined tags. We first find the popular tags that were assigned to at least 5K questions of SO. Then we find the questions that use one of these tags. We count the number of unanswered and answered questions connected to each topic. We then measure the ratio of unanswered and answered questions against each given topic. We then find the top 10 topics that were unanswered (e.g., jupyter-notebook, electron, rstudio) and another top 10 that were frequently answered (e.g., scheme, sql-server-2005, asp.net-mvc-2). Finally, we randomly choose 20 questions for each of these 20 topics, manually analyze a total of 400 questions (statistically significant with a confidence level of 95% and confidence interval of 5% [5]), and derive several qualitative insights.

Manual Analysis. Two of the authors conduct a qualitative study to see the different quality aspects of unanswered questions, contrasting them with answered questions. We analyze 400 questions by spending a total of 40 man-hours. We first examine the randomly sampled questions with no particular viewpoints in mind. We then discuss together to share our understanding and set a few dimensions to proceed with our analysis. Next, the sampled data is presented to each author for manual analysis. The authors rank them with four major quality aspects from 0 (lowest) to 5 (highest). The aspects are – i) problem description and code explanation ii) topic assignment ii) question formatting and iv) appropriateness of the code segments included with questions. Besides the above aspects, we also investigate whether the question was received an answer in a comment, redirected to another site (e.g., GitHub), asked suggestions, raised multiple issues in a single question, and marked as a possible duplicate. During the manual investigation, we periodically sat together and discussed the major inconsistencies in ranking until a consensus was reached. In particular, we resolved the inconsistencies by discussion when our ranking varied more than two in any aspect of a question.

Results. Fig. 6 shows the results of our manual analysis. For each of the four aspects, the average ranking for unanswered questions is lower than that of the answered questions. For example, the appropriateness of the code segments is 2.7 (on the scale of 5) for unanswered questions. Such rank is 3.63 for the answered questions. Our findings from the manual analysis are as follows.

• *Problem description and explanation of code segments*: We find that the problem description of the answered questions is complete, relevant, and easy to read. These questions include code segments when required and provide a useful explanation of the code elements so that users can comprehend them easily. When a question includes several code segments, we find a separate explanation for each of them. On the other hand, we find that the problem description is not sufficient and somewhat irrelevant in the unanswered questions. Questions without a clear problem statement are not useful to the readers. In the case of unanswered questions, we found long, redundant code segments that were not explained properly. They also do not provide any markers that could help one find the faulty part of the code. Sometimes the environment settings (e.g., operating system) and other technical details (e.g., software version) were found missing as well. In the quantitative analysis (section 3.1), we also find that the text-code ratio of the unanswered questions is lower than that of the answered ones. Thus, our manual investigation supports the quantitative difference in the text-code ratio between unanswered and answered questions.

• *Users experience and usage of appropriate terms.* We often see popular technical terms (e.g., GUI) in the texts of the answered questions which are familiar to the technical community. Such a use of popular words might improve the overall clarity of the question texts. On the contrary, we frequently notice the use of non-technical terms in the unanswered questions. This might occur due to the lack in experience (i.e., novice user) and domain expertise. In the quantitative analysis, we also find that (1) the submitters' reputation of the unanswered questions is lower than that of the submitters' of the answered questions on average, and (2) unanswered questions use more uncommon terms. Thus, our manual investigation agrees with the quantitative difference in the word uses between unanswered and answered questions.

• *Assignment of appropriate topics.* Tags assigned to the answered questions are precise and interesting enough to attract the experts of the domain. On the contrary, the tags assigned to the unanswered questions are generic (e.g., java). Our topic entropy shows that the tags of the unanswered questions are more generic, less precise than that of the answered questions. Users with low reputation (e.g., $reputation < 1500$) cannot add new tags to their questions [20] and are forced to use the existing generic or even potentially non-appropriate tags. Such a constraint might also prevent novice users of SO to assign proper tags to their questions.

3.3 Answering RQ3 : Construction of Prediction Models and Result Analysis

RQ1 and RQ2 investigate why the questions of SO remain unanswered using quantitative and qualitative approaches. Predicting a question (during its submission) that might not be answered could help one (1) improve the question and (2) thus receive the answers in time. We construct four machine learning models to predict whether a given question would be answered or not. In this section, we describe how we construct these models and evaluate their performance. We also compare our model performances with the baseline performances. Finally, we present a case study to demonstrate the generalizability of our models.

Algorithm Description. We use four algorithms for building prediction models – i) Decision Trees (CART) [7], ii) Random Forest (RF) [6], iii) K-Nearest Neighbors (KNN) [1], and iv) Artificial Neural Network (ANN) [33]. We choose these algorithms for two reasons. First, they are capable of building reliable models, even when the relationship between the predictors and class is nonlinear or complex. According to our comparative study, the relationship between question classes and their corresponding feature values might be complex. Second, these algorithms are widely used in the relevant studies [26, 28, 29]. Thus, we choose these four popular machine learning techniques that have different learning strategies to predict the unanswered questions. Moreover, they are non-parametric machine learning techniques. That is, they do not make any assumptions on the underlying data distribution.

Classification and Regression Tree (CART) and Random Forest (RF) are tree-based algorithms where the trees are constructed using predictors as non-leaf nodes and the classes as the leaf nodes. They show how individual features can affect the overall prediction. On the other hand, algorithms such as ANN and KNN combine all features (a.k.a., predictors) for determining the class label of an instance. ANN constructs a non-liner classification model during training, whereas KNN sorts the multi-dimensional feature space based on their classes.

Metrics of Success. We evaluate our prediction models using four appropriate performance metrics - *precision*, *recall*, *F1-score*, and *classification accuracy.*

Model Configuration. We use a grid search algorithm, namely *GridSearchCV*, from the scikit-learn library [24] to select the best model configuration. GridSearchCV algorithm works by running and evaluating the model performance of all possible combinations of parameters. We also examine the accuracy of both train and test datasets and adjust the parameters so that models do not over-fit. We use the following parameter settings to configure our models.

ANN is configured using two hidden layers with each having eight activation units, logistic *Sigmoid function* as the activation function, *adam* as the weight optimizer, learning rate of 0.0001 and a batch size of 200. For KNN, we apply the K-Dimensional Tree (KDTree) algorithm to compute the nearest neighbors between 50 neighbors where, distance is used for weight function. For DT, Gini is used as the function to measure the quality of split and two is used as the depth of the tree. For RF, the number of trees is used 5 and Gini is used for splitting criteria.

Dataset Selection. We keep the training and testing data separate. We use the attributes of the questions submitted on SO in the year 2016 or earlier for training and questions submitted in the year 2017 for the testing purpose. We have several time-dependent features (e.g., user reputation). Thus, we make sure that we predict the unseen questions based on past questions.

Prediction of Unanswered Questions. To see how well the prediction models perform based on our features, we conduct an experiment with our models. Table 3 shows the experimental results of our models.

Table 3: Experimental Results

Technique	Question	Precision	Recall	F1-Score	Accuracy
Decision Trees	Unanswered	72.57%	67.37%	69.87%	70.95%
	Answered	69.55%	74.54%	71.96%	
Random Forest	Unanswered	77.68%	80.83%	79.22%	78.80%
	Answered	80.02%	76.77%	78.36%	
ANN	Unanswered	58.59%	72.37%	64.75%	60.60%
	Answered	63.87%	48.84%	55.35%	
KNN	Unanswered	67.62%	67.46%	67.54%	67.57%
	Answered	67.54%	67.69%	67.62%	

We see that all four models can predict the question classes with 61%-79% accuracy. Our primary focus is to predict the unanswered questions. We thus analyze how the models perform to predict them. Random Forest performs the best among four models with precision 77.68% and recall 80.83%. Decision Tree is found as the second best with precision 72.57% and recall 67.37% followed by KNN with precision 67.62% and recall 67.46%, and ANN with precision 58.59% and recall 72.37%. One clear distinction can be drawn from the results that, individual feature processing algorithms such as Random Forest and Decision Tree are comparatively better suited for our dataset than the collective feature processing algorithms such as ANN and KNN. Moreover, Random Forest employs an ensemble learning, considers multiple trees to come to a decision and thus can avoid over-fitting and suited better for our dataset. Constructing multiple decision trees also helps the Random Forest model to capture non-linear relations between the predictors and classes more accurately for our problem. On the contrary, our dataset might not be well suited for ANN and KNN.

We investigate 40 cases (20 unanswered + 20 answered) where all of our models fail. Such insights might help and motivate future studies to improve models' performance. According to our investigation, our models misclassify the unanswered questions that are related to a few rising technologies (e.g., netflix zuul, libgdx). We see that the presentation quality, explanation of code segment, readability of text, and topic selection of these questions were reasonably well. Unfortunately, they might be remained unanswered due to the lack of experts. Our models often inaccurately predict the answered questions when they have long code with a more concise explanation. We use the text-code ratio instead of the length of the text to overcome such situations. However, our models fail to handle a few extreme cases [22]. Our analysis suggests that these questions received answers because either they included self-explanatory code segments or the code segments were capable of reproducing the issues reported at the questions [17].

Feature Strength Analysis. While answering RQ1, we showed that there are several attributes, whose values are different for unanswered and answered questions. Such findings indicate that these attributes might be the key estimators in predicting whether a question will be answered or not. A ranking of these attributes might help to select the top estimators in the prediction task. We thus use *ExtraTreesClassifier*, an inbuilt class of scikit-learn [24] to rank the features.

As shown in Fig. 7, topic response ratio has the highest classification strength. It indicates that selection of appropriate tags (or topics) during question submission at SO is important. According to our investigation, topics related to software libraries, frameworks

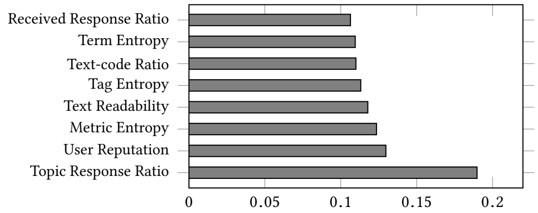


Figure 7: Feature Rank

are less likely to be answered in SO. User reputation is the second most effective feature. That is, the trusted users by the SO community are more likely to get answers in response to their questions. Reputation also serves as a proxy to a user's skills or expertise in a particular domain. The next two important features are metric entropy and text readability. They suggest that the use of terms and reading ease of the question texts is important to receive answers. The use of ambiguous complex terms can prevent a question from being answered. The remaining four features have low impacts on receiving answers.

Comparison with Baseline Models. According to the results (Table 3), our proposed models able to predict questions about 60–80% of the time correctly. However, we are interested to compare our models with the models studies previously. Thus, we choose two state-of-the-art studies – i) Saha et al. [29], and ii) Calefato et al. [9] related to our study. From the study of Saha et al. [29], we consider eight out of twelve features and discard the rest. We discard the four features such as – number of views, question score, number of favorite counts, and number of comments because they are not available at the submission time of a question. From the study of Calefato et al. [9], we consider all nine features. Next, we build four machine learning models using their attributes for each study. To create a level playing field, we also tune the parameters of baseline models to ensure the best performances. We finally compare the performance between baseline models and the proposed models. In particular, we compare the precision and overall accuracy between baseline models and our models.

Fig. 8 shows the comparative results between the proposed models and the baseline models. The performance of the models by Calefato et al. [9] is found comparatively poor than the others. Their accuracy and precision were found in the range of 50–52% for all the four algorithms. The strength of the features used by Saha et al. [29] is relatively higher than the features of Calefato et al. We see that the models trained with the features by Saha et al. outperform than that of the models by Calefato et al. The performance of the models by Saha et al. is about 1–7% higher than the models by Calefato et al. The overall accuracy of the models by Saha et al. is found 53–55% and the precision is found 53–59%. However, the proposed machine learning models outperform both the baseline models. The precision of the proposed models is about 6–25% higher than the models by Saha et al., and about 9–27% higher than the models by Calefato et al. Proposed models also have the improvement of overall accuracy between 5–25% from Saha et al., and 7–27% from Calefato et al.

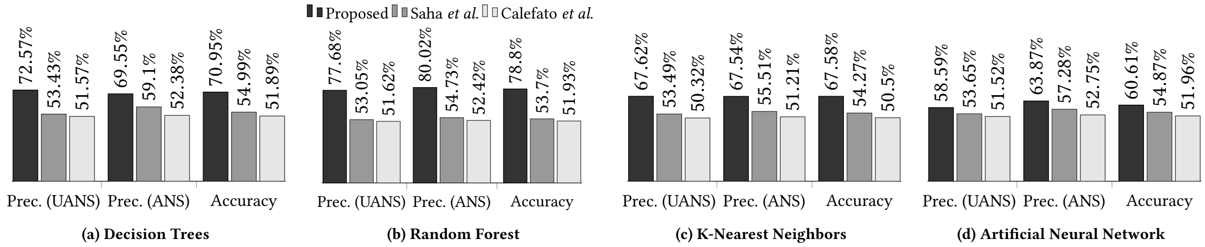
We further attempt to investigate why the baseline models fail that much in predicting unanswered questions. We manually investigate 50 cases where both the baseline models fail. Our analysis shows that predictions often go wrong due to the length of the questions. Both the baseline studies consider the character length of the questions (title + body + code segment). Their models often detect the questions having long length as answered. However, our analysis suggests that questions with short texts and long code segments are often remain unanswered. In our study, we use the text-code ratio that might not suffer as the length of the questions do. The presence of code also misleads the baseline models. The models are trained in a way that the questions with code segment are more likely to be answered. However, we find several incorrect cases where questions do not have any code segments, and yet they get answers. We also find that the attributes such as the number of tags, presence of external link, submission time have almost identical values for both the question classes. Thus the baseline models find difficulty while classifying questions based on these attributes.

Table 4: Model Performance with C++ and Go

Technique	Question	Precision		Accuracy	
		C++	Go	C++	Go
Decision Trees	***Unanswered***	66.02%	65.28%	67.40%	66.0%
	Answered	69.03%	66.74%		
Random Forest	***Unanswered***	64.10%	62.38%	66.3%	64.02%
	Answered	69.31%	66.17%		

Generalizability of the Models' Performance. In our study, we attempt to extract such features that are not dependent on any programming languages. Here, we test the models to confirm the generalizability of their performance and the strength of the features. We select questions from two different programming languages that are not used to train our models. In particular, we collect 100K questions (50K unanswered + 50K answered) of C++ programming language and 8K questions (4K unanswered + 4K answered) of Go programming languages. Go is an emerging programming language and thus we could not collect more questions. We select Random Forest and Decision Trees to test with C++ and Go since the experimental results (Table 3) show that they are well suited for our dataset and features.

The precision and accuracy of the models for C++ and go reported at Table 4 show that the models perform reasonably well, although they lose some precision and accuracy (less than 15%). Our further investigation reveals a few causes that might explain why the performance has declined. We see that the code segments included in the questions related to the Go language are comparatively small than that of the Java and C# languages. On the contrary, the length of C++ code segments are fairly long but self-explanatory. The users often add a brief description. Moreover, Go is an emerging language, and most of the questions discuss theory and do not include any code segments. These scenarios above might affect the attributes such as text-code ratio, text readability. Despite having such discrepancies, the proposed model can predict the unanswered question with a maximum of about 66% precision and 67% accuracy. Such findings confirm the general acceptance of our features and models in predicting the unanswered questions.



(a) Decision Trees **(b) Random Forest** **(c) K-Nearest Neighbors** **(d) Artificial Neural Network**

Figure 8: The performance of different algorithms applied to the raw feature values (UANS: Unanswered, ANS: Answered).

4 KEY FINDINGS & GUIDELINES

Our study provides several insights that explain why a question might fail to receive answers at SO. These insights may guide the users of SO, especially, novice users to improve their questions and increase the chance of getting answers.

(F1) Question format matters. Proper formatting of the description in a question is essential. In particular, the different elements (e.g., text, code, hyperlink) are recommended to place inside the appropriate HTML tags to improve their visibility and readability. Otherwise, the question might be hard to read, which could force the expert users to leave the questions without answering.

(F2) Attach code segment when required. If a question describes a code related issue, it should include an example code segment to support the problem statement. We find that users often request for code segments in the comment thread while trying to answer the code related questions. More importantly, one should include non-redundant code that is able to reproduce the reported issues in the question description. According to an earlier finding, in Stack Overflow, about 22% of the code segments fail to reproduce the reported issues in their questions [17]. In our manual analysis, we found that such lack of reproducibility could also leave a question unanswered.

(F3) Improper and non-technical description hurts. A clear description might prompt the expert users to answer a question. According to our investigation, we also find that the usage of appropriate technical terms is important. An ambiguous and non-technical description of a technical problem might mislead the users and thus hurt its chance of getting the solutions.

(F4) Stack Overflow is not suitable for all topics. Several topics are more likely to be unanswered in SO. According to our analysis, top unanswered topics are primarily related to programming IDEs, libraries, and frameworks. A lot of unanswered questions from these topics are related to internal configurations and algorithms. To answer such questions, the creator and the maintaining team need to be active users at SO which they are not. Hence, many unanswered questions are redirected to Github issues through comments [19] where the members from the development team are involved. In addition to that, questions related to the hardware configuration of network devices are also more likely to be unanswered. Such questions can be resolved by IT experts who are not that much active at SO. During manual analysis, we see that 8% of unanswered questions are directed to other sites. Thus, the users should (1) avoid submitting the off-topic questions and (2) carefully select the tags for their questions.

(F5) Multiple issues in a single question hurts. Some users often discuss multiple issues in a single question, which is not recommended by the community members. According to our investigation, question with multiple issues or concerns often fail to receive the answers.

(F6) Unanswered questions are not always really unanswered. We find that several questions received working solutions in their comment threads. According to our manual investigation, 5% unanswered questions receive answers in the comment threads. Thus, although many questions are seen as unanswered, they are actually answered.

(F7) Miscellaneous findings. We also find several other causes that might leave the questions unanswered. First, several questions remain unanswered since they discuss outdated technology. Second, many unanswered questions were found with misleading titles. Third, users sometimes requested for suggestions instead of discussing problems clearly and asking particular solutions. Forth, some questions were perceived as duplicates to other questions.

5 RELATED WORK

Several studies [3, 26, 27, 29] focus on classifying the SO questions. Saha et al. [29] introduce machine learning models to classify unanswered and answered questions of SO. Their models depend on the attributes (e.g., view count and scores) that are not available during the submission of questions. Thus, their models might fail to predict the unanswered questions during their submission. On the contrary, we investigate the question characteristics that are available during the submission of a question and construct prediction models to predict the unanswered questions. Saha et al. [29] also perform an initial investigation to understand why the questions of SO remain unanswered. They report that the lack of user interest is a major factor that might leave a question unanswered. We find that the user's interest often depends on the topics of the questions. The answering rate of several topics is fairly high, whereas several topics suffer from a low answering rate. We perform a comparative study on the question characteristics between answered and unanswered questions and report the top 10 topics (i.e., tags) which are more likely to remain unanswered.

Asaduzzaman et al. [3] conduct qualitative analysis to understand why questions of SO remain unanswered. They report several characteristics (e.g., impatient users) which are difficult to measure in practice. Besides, they do not provide any classification or prediction models for separating unanswered questions from the answered ones. Thus the strength of their features in classifying questions is not well understood. We automatically compute eight important features (e.g., topic response ratio) of SO questions and use them to predict the unanswered question. Furthermore, our prediction model outperforms two baseline models.

Calefato et al. [9] study a set of factors that are likely to influence the chance of getting answers by the questions at SO. They analyze about 87K questions and suggest that four factors might affect the chance of obtaining successful answers to the SO questions. They are presentation quality (e.g., presence of code snippets, question length), sentiment polarity (e.g., positive, negative), question posting time (e.g., day of week), and user reputation. According to our investigation, user reputation is one of the important attributes to predict unanswered questions. However, our feature set is more powerful for predicting the unanswered questions during their submission than those of Calefato et al. [9].

Ponzanelli et al. [26, 27] present an approach to classify the questions according to their quality (i.e., score). Our investigation reports that about 99% of questions having negative scores receive one or more answers. On the other hand, about 11% of questions which have positive score do not receive any answer [23]. Therefore, the classification model based on the score might not perform well in predicting the unanswered questions. Although a few of the features (e.g., metric entropy) are common between their model and ours, we extract more appropriate features to predict the unanswered questions.

Rahman and Roy [28] propose a model for predicting unresolved questions (i.e., no submitted answers were accepted as solutions by the questioner) of SO. Their feature analysis gives a key insight for understanding the difference between unresolved and resolved questions. We use a few of their features (e.g., topic entropy) in detecting unanswered questions. However, the remaining features (e.g., answer rejection ratio) are only effective in detecting the unresolved questions rather than the unanswered questions.

Chua and Banerjee [10] develop a conceptual framework known as the Quest-for-Answer to explain why some questions in Q&A sites draw answers while others remain ignored. They attempt to validate their framework empirically using 3000 questions (1500 unanswered + 1500 answered). However, some of their features are difficult to measure in practice (e.g., user politeness). A few of them (e.g., view count, question age) are also not available during the submission of questions. The remaining features (e.g., title length, description length) were employed by Calefato et al. [9]. However, our models outperform the models of Calefato et al. Wang et al. [32] investigate the factors for fast answers in technical Q&A sites. We not only investigate the causes of unanswered questions but also deliver machine learning models that can predict unanswered questions during their submission. To the best of our knowledge, no existing models are available to predict the unanswered questions during question submission time, which makes our work novel.

6 THREATS TO VALIDITY

This section discusses threats to the validity of this study. First, the baseline and the proposed models are built based on the four machine learning algorithms. One can argue that the results can vary with other algorithms. Saha et al. [29] use the Naive-Bayes algorithm in their study. However, the algorithm does not perform well in their study, and thus we discard this algorithm from our experiment. Furthermore, the algorithms used in our study are widely used by a number of related studies.

Second, we use five different text readability metrics in the proposed models. Other text readability metrics can have different impacts on the results. However, we also analyze the result with other metrics and find that our reported metrics perform better than them.

Third, during the manual investigation of answered and unanswered questions in RQ2 (Section 3.2), the participants remain aware of the status of the questions. One can argue that, if the analysis were performed keeping the participants blind, the bias could have been avoided. However, we also make one of the participants blind about the status of the questions and perform manual analysis on a sample set. We do not find any significant inconsistency between blind and non-blind observations. Thus, such a threat might have been mitigated.

Finally, to balance the answered and unanswered questions in the dataset, we select random samples from the answered questions and all from the unanswered questions. Statistical conclusion validity might arise due to undersampling the data [15]. However, we perform statistical tests routinely to justify our conclusions and thus avoid such threats.

7 CONCLUSION

A significant part of the questions of Stack Overflow do not receive any answers. Predicting them beforehand can help one improve the questions and thus receive the answers in time. In this paper, we contrast between unanswered and answered questions of SO quantitatively and qualitatively. We analyze 4.8 million questions and build models to predict the unanswered questions during their submission. Our contribution in this paper is threefold. First, we conduct a quantitative analysis and find that topics discussed in a question, the experience of the question submitter, and readability of question texts are likely to predict whether a question will be answered or not. Second, we manually investigate 400 questions (200 unanswered + 200 answered) and demonstrate that our qualitative findings support the quantitative findings. Our manual analysis also reveals several non-trivial insights (e.g., avoid multiple issues in a single question) that could guide novice users to improve their questions. Third, we develop four machine learning models to predict the unanswered questions. Our models can predict the unanswered questions during their submission time with about 78% precision and about 79% overall accuracy, which are significantly higher than that of two baseline models. The models also perform reasonably well when we test them with questions from two different programming languages (i.e., Go, C++). Such performance also suggests the robustness of our models and features.

In future, we plan to develop tool supports that could help the users revise their potentially unanswered questions.

ACKNOWLEDGMENTS

This research is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), and by a Canada First Research Excellence Fund (CFREF) grant coordinated by the Global Institute for Food Security (GIFS).

REFERENCES

[1] N. S. Altman. 1992. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician* 46, 3 (1992), 175–185.

[2] J. Anderson. 1983. LIX and RIX: Variations on a little-known readability index. *Journal of Reading* 26, 6 (1983), 490–496.

[3] M. Asaduzzaman, A. S. Mashiyat, C. K. Roy, and K. A. Schneider. 2013. Answering questions about unanswered questions of stack overflow. In *Proc. MSR*. IEEE Press, 97–100.

[4] Y. Benjamini and D. Yekutieli. 2001. The control of the false discovery rate in multiple testing under dependency. *The annals of statistics* 29, 4 (2001), 1165–1188.

[5] S. Boslaugh. 2012. *Statistics in a nutshell: A desktop quick reference.* "O'Reilly Media, Inc.".

[6] L. Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.

[7] L. Breiman. 2017. *Classification and regression trees.* Routledge.

[8] F. Calefato, F. Lanubile, M. C. Marasciulo, and N. Novielli. 2015. Mining successful answers in stack overflow. In *Proc. MSR*. IEEE Press, 430–433.

[9] F. Calefato, F. Lanubile, and N. Novielli. 2018. How to ask for technical help? Evidence-based guidelines for writing questions on Stack Overflow. *IST* 94 (2018), 186–207.

[10] A. Y. K Chua and S. Banerjee. 2015. Answers or no answers: Studying question answerability in stack overflow. *JIS* 41, 5 (2015), 720–731.

[11] M. Coleman and T. L. Liau. 1975. A computer readability formula designed for machine scoring. *Journal of Applied Psychology* 60, 2 (1975), 283.

[12] T. M. Cover and J. A. Thomas. 2012. *Elements of information theory.* John Wiley & Sons.

[13] Stack Exchange. Accessed on: December 2019. How does Reputation work? (Accessed on: December 2019). https://meta.stackexchange.com/questions/7237/how-does-reputation-work

[14] R. Gunning. 1952. The technique of clear writing. *Journal of writing* 1, 1 (1952), 1–50.

[15] M. Linares-Vásquez, G. Bavota, M. Di Penta, R. Oliveto, and D. Poshyvanyk. 2014. How do api changes trigger stack overflow discussions? a study on the android sdk. In *Proc. ICPC*.

[16] G. Harry M. Laughlin. 1969. SMOG grading-a new readability formula. *Journal of reading* 12, 8 (1969), 639–646.

[17] S. Mondal, M. M. Rahman, and C. K. Roy. 2019. Can issues reported at stack overflow questions be reproduced?: an exploratory study. In *Proceedings of the 16th International Conference on Mining Software Repositories*. 479–489.

[18] L. Moreno, J. J. Treadway, A. Marcus, and W. Shen. 2014. On the use of stack traces to improve text retrieval-based bug localization. In *Proc. ICSME*.

[19] Stack Overflow. Accessed on: December 2019. Cordova Media plugin doesn't work. (Accessed on: December 2019). https://stackoverflow.com/questions/38958605/cordova-media-plugin-doesnt-work

[20] Stack Overflow. Accessed on: December 2019. Create Tags. (Accessed on: December 2019). https://stackoverflow.com/help/privileges/create-tags

[21] Stack Overflow. Accessed on: December 2019. How do I ask a good question? (Accessed on: December 2019). https://stackoverflow.com/help/how-to-ask

[22] Stack Overflow. Accessed on: December 2019. wx.Frame error when calling one script from another. (Accessed on: December 2019). https://stackoverflow.com/questions/47331054/wx-frame-error-when-calling-one-script-from-another

[23] Stack Overflow. Accessed on: June 2019. StackExchage API. (Accessed on: June 2019). http://data.stackexchange.com/stackoverflow

[24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[25] E. Pitler and A. Nenkova. 2008. Revisiting readability: A unified framework for predicting text quality. In *Proc. EMNLP*.

[26] L. Ponzanelli, A. Mocci, A. Bacchelli, and M. Lanza. 2014. Understanding and classifying the quality of technical forum questions. In *2014 14th International Conference on Quality Software*. IEEE, 343–352.

[27] L. Ponzanelli, A. Mocci, A. Bacchelli, M. Lanza, and D. Fullerton. 2014. Improving low quality stack overflow post detection. In *2014 IEEE International Conference on Software Maintenance and Evolution*. IEEE, 541–544.

[28] M. M. Rahman and C. K. Roy. 2015. An insight into the unresolved questions at stack overflow. In *Proceedings of the 12th Working Conference on Mining Software Repositories*. IEEE Press, 426–429.

[29] R. K. Saha, A. K Saha, and D. E. Perry. 2013. Toward understanding the causes of unanswered questions in software information sites: a case study of stack overflow. In *Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering*. ACM, 663–666.

[30] E. A. Smith and R. J. Senter. 1967. Automated readability index. *AMRL-TR. Aerospace Medical Research Laboratories (US)* 1, 1 (1967), 1–14.

[31] M. Squire and C. Funkhouser. 2014. " A Bit of Code": How the Stack Overflow Community Creates Quality Postings. In *2014 47th Hawaii International Conference on System Sciences*. IEEE, 1425–1434.

[32] S. Wang, T.H. Chen, and A. E. Hassan. 2018. Understanding the factors for fast answers in technical Q&A websites. *ESE* 23, 3 (2018), 1552–1593.

[33] P. J. Werbos. 1990. Backpropagation through time: what it does and how to do it. *Proc. IEEE* 78, 10 (1990), 1550–1560.

[34] Wikipedia. Accessed on: April 2018. Readability. (Accessed on: April 2018). https://en.wikipedia.org/wiki/Readability

[35] H. Zhang, S. Wang, TH P. Chen, Y. Zou, and A. E. Hassan. 2019. An Empirical Study of Obsolete Answers on Stack Overflow. *TSE* 1, 1 (2019), 1–25.