Two Perspectives on Software Documentation Quality in Stack Overflow

Mathias Ellmann
University of Hamburg
Hamburg, Germany
mathias.ellmann@uni-hamburg.de

Marko Schnecke
Hamburg, Germany
marko.schnecke@mailbox.org

ABSTRACT

This paper studies the software documentation quality in Stack Overflow from two perspectives: the questioners' who are accepting answers and the community's who is voting for answers. We show what developers can do to increase the chance that their questions or answers get accepted by the community or by the questioners. We found different expectations of what information such as code or images should be included in a question or an answer. We evaluated six different quality indicators (such as Flesch Reading Ease or images) which a developer should consider before posting a question and an answer. In addition, we found different quality indicators for different types of questions, in particular error, discrepancy, and how-to questions. Finally we use a supervised machine-learning algorithm to predict when an answer will be accepted or voted.

CCS CONCEPTS

• **General and reference → Computing standards, RFCs and guidelines**; **Empirical studies**; *Metrics*; *Evaluation*; Measurement; Validation;

KEYWORDS

Software Analytics, Stack Overflow, Software Documentation, User Analysis, Knowledge Sharing

ACM Reference Format:
Mathias Ellmann and Marko Schnecke. 2018. Two Perspectives on Software Documentation Quality in Stack Overflow. In *Proceedings of the 4th ACM SIGSOFT International Workshop on NLP for Software Engineering (NL4SE '18), November 4, 2018, Lake Buena Vista, FL, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3283812.3283816

1 INTRODUCTION

Over 4.8 million Stack Overflow members posted over 10,898,656 questions[1] by 2016 and received an answer within eight minutes by the software community. About three million[2] of these questions do not meet the quality requirements of the questioners or the software community as being unclear, incomplete or imprecise [2] and were edited by about 90,000 software developers in Stack Overflow (SO) [9].

Stack Overflow members can ask questions and accept an answer without any needed privileges[3]. Other privileges are given after they have successfully contributed content to Stack Overflow. Questioners can set an accepted flag after they have decided on an appropriate answer to clarify that the answer has helped them to solve their software problem and has meet their quality expectations. Stack Overflow members can vote for the best answers they think serves best to enrich the community knowledge with high quality content.

Questions can be distinguished between 10 question types [10] as How-To or Error questions. How-To questions often ask for a step-to-step instruction e.g., how to change the color of a list element in the User Interface of a mobile application. Error questions on the other hand ask for a solution to fix a specific Error that occurs e.g., by using an IDE and its software components. Questions and answers in Stack Overflow can contain different information artifacts as text, images or code to support the overall understanding of the questions and answers. The quantity of those artifacts can vary within the 10 question types as Error questions might include a higher quantity of code snippets than other questions.

In this paper we evaluate randomly chosen questions as well as different kinds of questions as How-To or Error questions [10] and answers being voted or accepted by different parties in Stack Overflow and discuss the consequences for the documentation quality by adding or leaving information artifacts as code example or images. Finally, we evaluate with a Naïve Bayes machine learning algorithm the optimal combination of artifacts to meet the expectations of questioners or the software community in Stack Overflow.

2 RELATED WORK

A recent study of Calefato et al. [3, 4] evaluated how Stack Overflow users can increase the chance to receive an accepted answer. The researchers found that answerers are more successful if they are polite to the questioner and that using a friendly tone when answering overall encourages new users to participate more in a discussion. Also Novielli et. al. [6] found that sentiments might play an important role when exchanging information about technical issues in Stack Overflow. Calefato et al. also found, that the user reputation positively influences the success of a question. They defined answers that are chosen as accepted by the questioner to be considered as "successful". Our approach differentiates between

[1] http://stackoverflow.com/questions as of January 2016
[2] http://stackoverflow.com/unanswered as of January 2016
[3] http://stackoverflow.com/help/privileges

NL4SE '18, November 4, 2018, Lake Buena Vista, FL, USA

ACM ISBN 978-1-4503-6055-5/18/11…$15.00
https://doi.org/10.1145/3283812.3283816

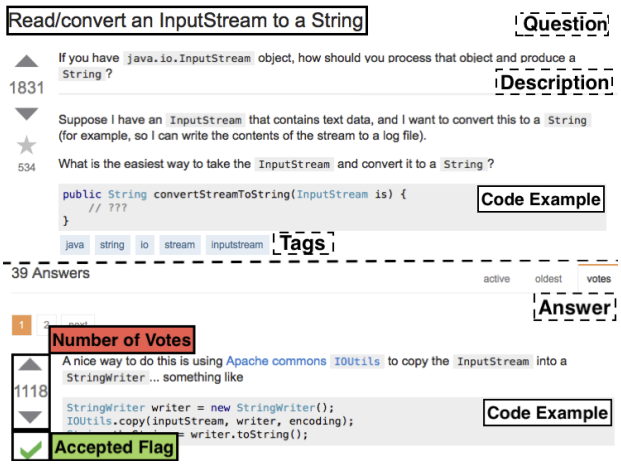

Figure 1: Example of a SO post: question, description, tags and a voted as well the accepted answer - Question Id: 309424

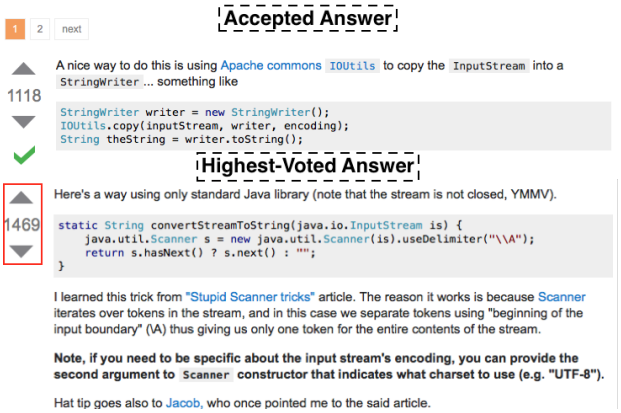

Figure 2: Example of a valuation difference between the accepted answer by the questioner and the voted answer by the SO community - Question Id: 309424

two different actions made by questions and the software community that can lead to a success of questions and answers in Stack Overflow.

Treude et al. [10] as well as Wang et al. [11] found different topics are discussed in SO questions. Treude et al. differentiate between 10 different question types with success rates from 33% to 92%. Based on this categorization of questions, we evaluate whether the different question types benefit from a different quantity of information artifacts as codes snippets etc. According to their definition, "a successful question has an accepted answer, and an unsuccessful question has no answer". In this study we distinguish between three different question types as Error, Discrepancy and How-To questions and evaluate the factors of those questions that make them successful.

Ravi et al. [8] studied great questions in Stack Overflow and Ponzanelli et al. [7] combined the quality filters of Stack Overflow with a popularity metric (taking the history of the questioner into account) and a readability metric (measuring how easy it is to read the question). In our research we use metrics that evaluate the existence or quantity of different information artifacts and readability metrics to train a machine-learning algorithm that can predict its success.

3 RESEARCH DATA AND METHOD

Q&A websites like Stack Overflow are one of the most important knowledge resources [5] for software developers. Stack Overflow and all other 148 Q&A websites that are related to Stack Exchange, stems from its community resources and their individual and collective assessment of Q&As in Stack Overflow. Often the community members in Stack Overflow have different reasons why they accept or vote a question leading to uncertainty when to perform a specific action[4] to support the community. In this paper we will empirically investigate when those actions will be taken and how a Stack Overflow member can contribute highly valuable content to the platform.

Figure 1 shows a Stack Overflow question including both actions as voting or accepting an answer to evaluate the content of the Stack Overflow website. A Stack Overflow post contains a question, tags and an answer that includes different information artifacts such as code, images, listings or other. The questioner can accept an answer that is displayed to other community members as an accepted flag (Figure 1) - accepted answers are always displayed first. Despite that the software community can vote for an answer being mostly valuable for the software community. The answer that is accepted is not necessarily the best answer from the software community's point of view in terms of quality as a comparison of the votes scores of Figure 2 shows. Often very active community members do not set an accepted flag despite the fact that the community has highly rated an answer. 18.45% (58,293) of the accepted answers do not have the highest vote-score by considering only answers that have a competing answer. In this paper we will answer the following questions.

(1) What information artifacts do questioners and the software community expect in questions and answers?
(2) What combination of information artifacts works best to fulfill the quality expectation of questioners and the software community?

We downloaded the Stack Overflow data dump located at archive.org[5] on May 2014. The data consists of over 7.2 million questions and over 12 million answers. Each question has on average 1.75 answers. Stack Overflow members provided at most 518 answers for a question. We extracted all information artifacts such as the code, images, etc. from the questions and answers. We did this for all questions and for specific question types such as Error, Discrepancy and How-To [10] questions. Error questions are questions that contain a concrete error message. We found 152,233 Error questions by filtering our data sample by the keyword "error" and rechecked the sample by randomly choosing 20 questions of which 18 were actually Error questions. Discrepancy questions are questions that regard unexpected behavior. Because most questioners ask why the

[4] http://meta.stackexchange.com/questions/5234/how-does-accepting-an-answer-work

[5] https://archive.org/details/stackexchange

Table 1: Best number of information artifacts to increase significantly (> average) the accepted or voted answer score

	All Questions (1,000,000)		Error Questions (152,233)		Discrepancy Questions (50,787)		How-To Questions (302,498)	
Metric / Answer Type	Accepted Answer	Voted Answer	Accepted Answer	Voted Answer	Accepted Answer	Voted Answer	Accepted Answer	Voted Answer
Question (Words)	>2	6-8	>2	>3	Not Significant	Not Significant	>2	5-8
Description (Words)	0-50	0-50	50-75	0-50	0-50	0-50	150-200	0-50
Description Flesch Reading Ease	60-120	>80	60-100	100-120	80-100	100-120	60-120	100-120
Description Code To Text Ratio (%)	1-70	1-40	10-80	1-40	10-70	1-30	1-70	1-30
Description Images	positive	negative	positive	negative	neutral	neutral	positive	negative
Description Listing	positive	positive	positive	positive	neutral	positive	neutral	neutral
Description Quotes	neutral	positive	neutral	positive	positive	positive	neutral	positive
Tags (Words)	2-5	2	3-4	Not Significant	2	Not Significant	2-4	Not Significant
Answer (Words)	>50	>75	>50	>75	>75	>100	>50	>100
Answer Flesch Reading Ease	40-80	20-60	40-80	40-60	40-80	40-60	40-80	40-60
Answer Code To Text Ratio (%)	1-90	1-60	1-90	1-60	1-70	1-30	1-90	1-70
Answer Images	positive							
Answer Listing	positive							
Answer Quotes	positive							

unexpected behavior occurs, we filtered all questions that contain the word "why" but do no include "error", "want" or "how-to" to find 50,787 Discrepancy questions. In a manual check 16 out of 20 questions actually were Discrepancy questions. How-To questions are questions that ask for instructions. We found 302,498 How-To questions by filtering for the keywords "how-to", "want" and "?" and then cleaned the resulting data by filtering out possible Decision, Error or Discrepancy questions. In a manual check we found that 16 out of 20 questions actually were How-To questions. Finally, we evaluated that a combination of features might lead to a specific outcome as a question or an answer is accepted or voted. To achieve that we used the Naïve Bayes machine learning algorithm[6].

4 EVALUATION CRITERIA OF THE ANSWERS

Every member in Stack Overflow can ask a question and accept an answer in order to show other members the answer has helped him. From a sample of 1,000,000 questions with 1,748,284 answers we found that only 58.36% of the Stack Overflow questioners accepted an answer (58,17% of the questions have an accepted answer). This might related to some of the questions not being answered (11.03%), or answers being unsuitable (30.80%). 246,043 (7.99%) of the questioners are very active on Stack Overflow and provided more than the average of 4.07 answers (median 0). We randomly chose 20 questions and found that questioners who are very active (answer rate > 4.07) mostly accept an answer because it is most comprehensible (Flesch Reading Ease: 69.06) or has a list of instructions, particularly in How-To questions.

In addition to an individual assessment of an answer, the Stack Overflow community can also vote for an answer as being most helpful for the proposed problem. The average vote score for answers is 2.72, which shows that Stack Overflow members act collectively within the Stack Overflow platform, to spare time for other members. A first impression of 20 Stack Overflow posts that have a very high number of answers (above the 1.75 average) and votes (above the 2.72 average) shows that the software community often up-votes answers that contain (a fraction of) information artifacts such as code examples or listings (p-value: 9,94E-093; +2.18 of the vote score) compared to others that did not. Often the number of comments of an answer that are attached can also be a driving factor to up-vote an answer. During the analysis of the 943,172 answers from the answer data sample, we found that while the average number of comments for all answers is 1,37 (median 0), the average number of comments for accepted answers is 2,04 (median 1) and 1,61 (median 1) for answers with a vote-score over 0. Comments can supplement an answer with additional links to external resources or useful explanations.

We observed that the questioners and the software community often have different opinions as to whether an answer is the best one out from several answers. From a sample of 988,255 posts, 182,333 (18.45%) of the accepted answers have not received the highest vote score. This can occur because the questioners have accepted the answer that came in first or because it is distinct from others. From a qualitative analysis of 20 accepted answers that were accepted after at least two answers were submitted, we found that questioners often accept answers that include a code example. We found that that community values answers that includes background information the developer has used e.g., provided hyper-links to external resources. By using our sample of 182,333 accepted answers, we found that answers are more likely to get accepted if they include an image (p-value: 2,61E-166, +15.88%), a quote (p-value: $\lim_{p\to 0}$,+10.14%), listings (p-value: $\lim_{p\to 0}$,+9.33%) or a URL (p-value: $\lim_{p\to 0}$, +4.21%). The average vote-score of these answers also increases (image p-value: 4,35E-27, +3.60; listing p-value: 9,94E-93, +2.17; quote p-value: 6,26E-74, +1.92 and url p-value: 1,64E-70, +0.61). We believe that the acceptance of an answer might also depend on the context and current activity of the developer who asks a question e.g., using an IDE and developing for Python or learning about a software product. We found that developers often post a code example in their question descriptions, therefore we believe the developer is developing (context: IDE, OS) and editing code (activity) at that moment. In a data sample of 581,627 questions that have an accepted answer, we found that questioners who provided a code snippet accepted an answer that also included a code snippet 82.12% (p-value: $\lim_{p\to 0}$) of the time; questioners who did not include code in their question only accepted an answer with code 47.42% (p-value: $\lim_{p\to 0}$) of the time. The same is valid for the usage of URLs; questioners who might use their web browser, copy and provide an URL in their question chose an answer that also included a URL in 45,49% (p-value: $\lim_{p\to 0}$) of the cases, while

[6] http://www.nltk.org/_modules/nltk/classify/naivebayes.html

questioners who did not use a URL chose in only 34.39% (p-value: 3,87E-86) of the cases an answer with a URL as accepted.

4.1 Relevant information artifacts in a Q&A

In Table 1 we calculated different metrics from the information artifacts, as a question, question description, tags or answer in Stack Overflow. For every metric we measured a significant increase above average (p-value < 0.05) of the accepted answer ratio and of the votes given by the software community.

We found that the question does not have to be very extensive to get an accepted answer. A more relevant information entity is the question description which often describes the overall problem in more detail. Therefore the question might only extend the list of tags given by the questioner. While the software community prefers short questions (<50 words) and long answers (>75 words), the questioners are content with slightly shorter answers (>50 words). We believe that the reason is that the software community values extra information for future use while the questioners are content with shorter answers, as long as they are substantial enough to enable them to continue their work. Questioners and the community expect, that answers are more complex and include elements such as terminologies, function names and concepts which is emphasized by an ideal Flesch Reading Ease score of 60-80. Especially the community seeks for questions being well explained that e.g., includes information about the used framework or the modified UI element.

Images, listings and quotes can often help to understand and visualize the overall problem respectively solution. We found that questions which include images or listings are more likely to receive an acceptable answer and that questions which include listings or quotes are more likely to get up-voted answers. Answers benefit very much from images, listings and quotes from both, the questioner and the community perspective. An answer arises in Stack Overflow after twelve minutes [1] throughout all posts respectively 8 minutes (median) in our data sample of 1,000,000 posts. It appears that images help the developer during his solution finding process especially when the answer arises within a short period of time. For Error questions, a 10% higher code-to-text ratio and a longer question description (50 - 75 words) increases the chance of receiving an accepted answer. That might be related to the current environment or the libraries used. Discrepancy questions on the other hand benefit from slightly longer answers and How-To questions receive better-voted answers if they use about 10% less code.

4.2 Prediction of accepted and voted answers

Most of the time a combination of information artifacts is needed to provide a successful question or answer. We trained a Naïve Bayes algorithm with 6 features (Flesh-Reading-Ease score, Code-to-Text ratio, word count, image exists, listing exists, quote exists) for answers and one additional feature for questions (number of tags). In a sample of 300,000, 50% of the questions had a vote-score above the average of 3.01 and 50% below. We tested the classifier with 15,000 randomly chosen questions of both categories. The accuracy of identifying a high quality question is 57.48%, the precision 57.13% and recall 59.94% in identifying a highly valuable question for the community. The most informative features are the Flesch Reading Ease (>100) and a Code-to-Text Ratio (0.05-0.25) for high quality questions. Low quality questions have a Flesch Reading Ease below 20. It shows that they are very difficult to understand. The community also sees a Code-To-Text Ratio > 0.75 as very negative. To evaluate high quality from the questioners point of view we used a sample of 600,000 questions, 50% with an accepted answer and 50% without. The accuracy is 55.35%, the precision is 55.08% and the recall is 58%. We did the same for the answers and could identify high quality answers evaluated by the software community with a 62.51% accuracy, 62.80% precision and 57.96% recall. In this case however, the most informative features for high quality answers are the number of words (> 200) and the existence of an image, quotes or listings. The accuracy to get an answer accepted is 61.82%, the precision is 62.80% and the recall is 57.96%. The code-to-text ratio plays a less significant role to get the answer not accepted.

5 CONCLUSION

In this paper we have shown the discrepancy in quality assessments of Q&As between questioners and the software community. We described what combination of information artifacts might lead to a specific outcome for all questions and for three different types of questions in Stack Overflow. We found that the expected information in an answer might also depend on the context of the developer whether he is programming e.g., by using an IDE or by using a web browser and searching for software development knowledge, which needs further investigation. Our results might help questioners and answerers to prepare the right amount of information artifacts in a Stack Overflow post before posting a specific question on Stack Overflow. This might help to reduce the effort the Stack Overflow community has to spend on editing questions and answers.

REFERENCES

[1] Ashton Anderson, Daniel Huttenlocher, Jon Kleinberg, and Jure Leskovec. 2012. Discovering value from community activity on focused question answering sites: a case study of stack overflow. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 850–858.

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

[3] Fabio Calefato, Filippo Lanubile, Maria Concetta Marasciulo, and Nicole Novielli. 2015. Mining Successful Answers in Stack Overflow. In *Proceedings of the 12th Working Conference on Mining Software Repositories*.

[4] Fabio Calefato, Filippo Lanubile, and Nicole Novielli. 2018. How to ask for technical help? Evidence-based guidelines for writing questions on Stack Overflow. *Information and Software Technology* 94 (2018), 186–207.

[5] Roehm Tobias Koschke Rainer Maalej Walid, Tiarks Rebecca. 2014. On the Comprehension of Program Comprehension. *ACM Transactions in Software Engineering and Methodology* 23, 4 (2014), 31:1–31:37.

[6] Nicole Novielli, Fabio Calefato, and Filippo Lanubile. 2015. The challenges of sentiment detection in the social programmer ecosystem. In *Proceedings of the 7th International Workshop on Social Software Engineering*. ACM, 33–40.

[7] Luca Ponzanelli, Andrea Mocci, Alberto Bacchelli, Michele Lanza, and David Fullerton. 2014. Improving low quality stack overflow post detection. In *Software Maintenance and Evolution (ICSME), 2014 IEEE International Conference on*. IEEE, 541–544.

[8] Sujith Ravi, Bo Pang, Vibhor Rastogi, and Ravi Kumar. 2014. Great Question! Question Quality in Community Q&A. *ICWSM* 14 (2014), 426–435.

[9] Marko Schnecke. 2015. *An empirical study to improve Q&As in Stack Overflow*. Technical Report.

[10] Christoph Treude, Ohad Barzilay, and Margaret-Anne Storey. 2011. How do programmers ask and answer questions on the web?: Nier track. In *Software Engineering (ICSE), 2011 33rd International Conference on*. IEEE, 804–807.

[11] Shaowei Wang, David Lo, and Lingxiao Jiang. 2013. An Empirical Study on Developer Interactions in StackOverflow. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing (SAC '13)*. ACM, 1019–1024.