2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR)

Can Duplicate Questions on Stack Overflow Benefit the Software Development Community?

Durham Abric, Oliver E. Clark, Matthew Caminiti, Keheliya Gallaba, and Shane McIntosh
Department of Electrical and Computer Engineering
McGill University, Montréal, Canada
[durham.abric, oliver.clark2, matthew.caminiti, keheliya.gallaba]@mail.mcgill.ca, shane.mcintosh@mcgill.ca

***Abstract*—Duplicate questions on Stack Overflow are questions that are flagged as being conceptually equivalent to a previously posted question. Stack Overflow suggests that duplicate questions should not be discussed by users, but rather that attention should be redirected to their previously posted counterparts. Roughly 53% of closed Stack Overflow posts are closed due to duplication. Despite their supposed overlapping content, user activity suggests duplicates may generate additional or superior answers. Approximately 9% of duplicates receive more views than their original counterparts despite being closed.**

In this paper, we analyze duplicate questions from two perspectives. First, we analyze the experience of those who post duplicates using activity and reputation-based heuristics. Second, we compare the content of duplicates both in terms of their questions and answers to determine the degree of similarity between each duplicate pair. Through analysis of the MSR challenge dataset, we find that although duplicate questions are more likely to be created by inexperienced users, they often receive dissimilar answers to their original counterparts. Indeed, supplementary textual analysis using Natural Language Processing (NLP) techniques suggests duplicate questions provide additional information about the underlying concepts being discussed. We recommend that the Stack Overflow's duplication policy be revised to account for the benefits that leaving duplicate questions open may have for the developer community.

I. Introduction

Stack Overflow is a popular Q&A forum for developers of all experience levels. Developers who congregate there can discuss and share knowledge about programming. As of February 2019, Stack Overflow has over 10 million registered users, over 17 million questions, and over 26 million answers.[1] Due to its popularity, many previous studies have investigated Stack Overflow and how developers interact with it. For example, past studies have explored user personality traits [7], [14], topic trends [4], [6], and leveraging crowd-curated knowledge for use in the IDE [3], [15].

Stack Overflow posts consist of exactly one question and a set of answers. A post can be open (indicating that an acceptable answer is still being sought), closed (indicating that an acceptable answer is no longer required), or locked (indicating that changes to the post are prohibited). Additionally, posts can contain URL links to other related posts.

While a post may be closed for quality reasons (e.g., off-topic question, scope too broad), the most frequently occurring closure reason is duplication. Recent studies [10] have shown that the proportion of posts closed due to duplication is increasing over time. A post is closed as a duplicate when it is deemed too similar to a pre-existing post. In this paper, we refer to the pre-existing post as the duplicate's *root* post. Additionally, we refer to any non-duplicate post as an *original* post. A duplicate post contains a link to its root post to direct users to the pre-existing version of the question and its answers (see our online appendix for examples[2]).

While duplication has been explored in other software development contexts (e.g., source code [12], bug reports [1], documentation [13]), little is known about the value that duplicate posts have on Stack Overflow. Most prior work on duplicates in Stack Overflow has focused on accurately predicting duplicate posts to ease the manual burden of duplicate detection [2], [11], [16] A particular inspiration for our work is that of Bettenburg et al. [8], who showed that while users who post duplicate bug reports are often stigmatized, those duplicates often provide useful additional information. Duplicate posts on Stack Overflow have a similar stigma associated with them.[3]

Similar to Bettenburg et al.'s observations on bug reports, we believe that a deeper analysis of duplicates on Stack Overflow will provide insight for users, moderators, and builders of Q&A sites. Indeed, we conjecture that duplicate posts on Stack Overflow are valuable to the developer community. First, duplicates provide an additional phrasing of a problem or solution that may help community members find it. Second, the additional answers that duplicates contain may prove more understandable to certain Stack Overflow users.

In this paper, we study duplicate questions on Stack Overflow in terms of both content and the users who create them. We quantify the experience of users who ask duplicate questions. Further, we measure the similarity between duplicate and root questions and their associated sets of answers. Through analysis of the MSR Challenge dataset [5], we address the following research questions:

(RQ1) Is user experience related to the likelihood of asking a duplicate question? The reputation (a measure of seniority for Stack Overflow users) of the asker is significantly different in the duplicate and original questions ($p < 0.05$,

[1] https://stackexchange.com/sites#users

[2] https://github.com/software-rebels/msrchallenge19/wiki

[3] https://meta.stackexchange.com/questions/10841/how-should-duplicate-questions-be-handled

2574-3864/19/$31.00 ©2019 IEEE
DOI 10.1109/MSR.2019.00046

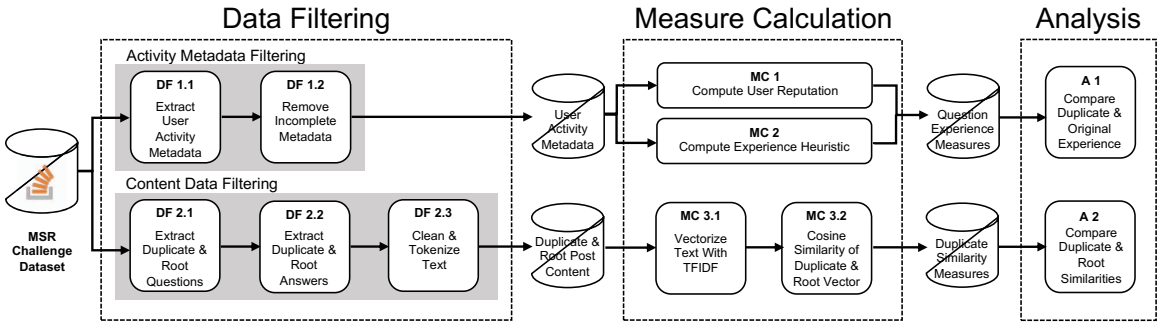


Fig. 1: An overview of our approach to study the MSR Challenge dataset

two-tailed Mann-Whitney U test). However, the magnitude of the difference is negligible (Cliff's delta = 0.019). On the other hand, the number of questions that users have previously asked is significantly different ($p < 0.05$, two-tailed Mann-Whitney U test) with a moderate effect size (Cliff's delta = 0.25).

(RQ2) Do duplicate posts contain unique information? The mean cosine similarity of duplicate questions (0.204) and answers (0.234) suggests that duplicate posts contain considerable degrees of both overlapping and unique information. Yet we find that question similarity does not imply answer similarity, suggesting that the even near-identical questions can still yield unique answers.

Replication. To facilitate future work, we have made a replication package publicly available online.[4]

II. Study Design

Figure 1 provides an overview of the approach that we followed to answer the research questions.

A. Data Filtering

Since some records are unsuitable for analysis, we first extract and preprocess data from the MSR challenge dataset.

DF 1.1: Extract User Activity Metadata. We retrieve *Id*, *OwnerUserId*, and *CreationDate* for all questions *(PostTypeId = 2)* and the *ParentId* for all answers *(PostTypeId = 1)* from the SOTorrent *Posts* table. User reputation and experience measures are determined by a user's prior activity on Stack Overflow; however, these values change over time. To reconstruct these measures at the time of a post, we extract data from the *Votes* and *PostHistory* tables. The Votes table provides upvotes & downvotes, accepted answer decisions, spam/offensive complaints, and bounties, while the *PostHistory* table provides suggested edit approval.

DF 1.2: Remove Incomplete Metadata. All anonymous user activity (i.e., those that lack an associated *UserId*) could not be used in calculating user experience. Moreover, all user activity with missing key values (e.g. *BountyAmount* on a *BountyStart* vote) could not be factored into reputation calculations. Thus, we filter out such entries during this step.

DF 2.1: Extract Duplicate & Root Questions. To extract the duplicate and root question posts, we first select all *PostId* and *RelatedPostId* pairs from the *PostLinks* table where *LinkTypeId = 3*. These values identify all duplicate and root post pairings. Next, we use the previously extracted set of *PostId* and *RelatedPostId* values (our candidate set) to filter through the *Posts* table. We select only the *Posts* that have a *PostId* included in the candidate set. All that is selected from the *Posts* table is the *PostId* and *Body* of the question. This step results in 647,664 candidate question posts, of which 31,111 are both a root and a duplicate question.

DF 2.2: Extract Duplicate & Root Answers. We select the corresponding set of answers for each candidate question. Treating the output from *DF1.1* as a set of candidate questions, we select all answers from *Posts* where *PostTypeId = 2* and *ParentId* belongs to the candidate set. We save only the *PostId*, *Body*, and *ParentId* fields of the candidate answer posts.

DF 2.3: Clean & Tokenize Text. Since the content of questions and answers are rendered as HTML, we remove all HTML tags from the content, leaving the raw text of all pairs of duplicate questions and answers. Question text is treated as its own document, while answer text is grouped by its parent question. Next, we tokenize the text, remove stop words, and apply the Porter stemmer to each surviving token using the Python Natural Language Tool Kit (NLTK) library [9].

B. Measure Calculation

Below, we explain the steps to compute our measures.

MC 1: User Reputation Measure. Stack Overflow uses a reputation heuristic to motivate the community of users to engage with the platform in constructive ways.[5] We use the official Stack Overflow reputation formula to recover the reputation of an asker at the time of a post. Unfortunately, the -1 reputation penalty issued for downvoting an answer and the site association bonus $+100$ reputation on registration could not be factored into our calculation, as that data is omitted from the MSR Challenge dataset to protect user anonymity.

[4]https://github.com/software-rebels/msrchallenge19

[5]https://meta.stackexchange.com/questions/7237/how-does-reputation-work

Nonetheless, the reputation of a user U at post time T is:

$$\begin{aligned} \textit{Reputation}(U,T) = 1 + [&5 \times \textit{QuestionUp}(U, t < T) \\ &+ 10 \times \textit{AnswerUp}(U, t < T) \\ &+ 2 \times \textit{EditAccepted}(U, t < T) \\ &- 2 \times \textit{QuestionDown}(U, t < T) \\ &- 2 \times \textit{AnswerDown}(U, t < T)] \\ &+ 2 \times \textit{AcceptAnswer}(U, t < T) \\ &+ 15 \times \textit{AnswerAccepted}(U, t < T) \\ &+ \sum_{\forall b \in \textit{BountyReceived}(U, t < T)} b_{\textit{amount}} \\ &- \sum_{\forall b \in \textit{BountyOffered}(U, t < T)} b_{\textit{amount}} \\ &- 100 \times \textit{Posts}(U, \textit{OffensiveFlags} > 5, t < T) \end{aligned}$$

All users start with a reputation of one, and reputation can never drop below that. Additionally, users can receive a lifetime maximum of 1,000 points for having edits accepted, and a daily maximum of 200 points from votes and edits.

MC 2: User Experience Heuristic. We use a heuristic to estimate user experience at the time a question was posted. To do so, we count the number of posts created by the user prior to posting the question under analysis.

MC 3.1: Vectorize Text with TFIDF. To calculate the similarity of text data, we produce vectors for all duplicate and root pairs (grouping answers by question). We use *Term Frequency-Inverse Document Frequency* (TFIDF) to weigh the vectors of terms in each document. Terms that rarely appear in the full corpus and/or frequently appear in the document will have a higher TFIDF weight than those appearing more frequently in the full corpus.

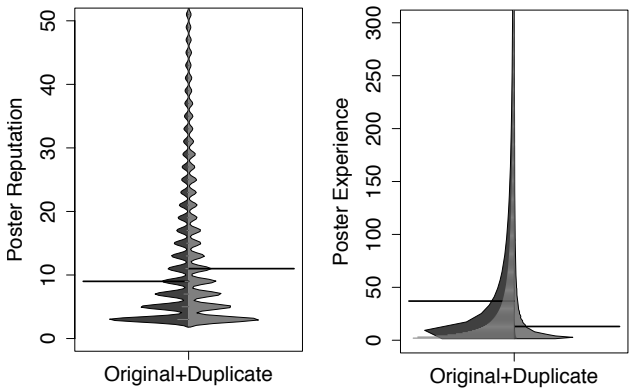
MC 3.2: Cosine Similarity of Duplicate & Root Vector. The similarity of our TFIDF vectors was assessed with the cosine similarity metric. Cosine similarity measures the difference in direction of two non-zero vectors. The cosine similarity produces a value between zero and one; one being for perfectly identical vectors, and zero being entirely unrelated vectors.

C. Analysis

The computed measures enable the following two analyses.

A 1: Compare Duplicate & Original Experience. The reputation and experience values for duplicate and original question askers were compared using the Mann-Whitney U-test—a non-parametric test that indicates the likelihood that two samples are drawn from the same population. Moreover, we use Cliff's delta to measure the practical difference between reputation and experience values of duplicate and original question askers.

A 2: Compare Duplicate & Root Similarity. To contextualize the cosine similarities of duplicates, random sets of questions and answers were processed using the same steps and compared to determine a baseline similarity for questions and answers without any obvious relation. To do so, we process sets of questions and answers with matching tags and compare them to the duplicate similarities. We then report observations by plotting the distributions and trends of similarity values.



(a) Reputation Measure (b) Experience Heuristic

Fig. 2: Experience measure distributions of question askers

III. Study Results

In this section, we discuss the results of our study with respect to the two research questions. For each research question, we first present the approach to addressing it and then discuss the results that we observe.

RQ1: Is user experience related to the likelihood of asking a duplicate question?

Approach. To determine the relationship between user experience and the creation of duplicate questions, we apply the heuristics discussed in *Section II.A.MC 1* and *MC 2*. Next, we apply the analysis techniques outlined in *A 1* to the experience measures of duplicate and original questions.

Observations. Analysis of duplicate and original questions shows that the askers of both question types have a median reputation of 5. The low median reputation values are influenced by the large number of questions asked by 'one-off' users that post a question as soon they create an account. Nevertheless, a Mann-Whitney U-test indicates that the reputation of askers of original questions is significantly larger than that of duplicates ($p = 7.93 \times 10^{-3}$). However, the Cliff's delta indicates that the practical difference is negligible ($\textit{delta} = 0.019$).

Figure 2a shows the distribution of reputation values for original and duplicate questions. Visual inspection of Figure 2a supports the observation that a user's reputation has little impact on whether they will ask a duplicate question. This observation may be influenced by users who use Stack Overflow passively and interact with other's posts, but rarely create their own.

The median-experienced asker of a duplicate question had previously asked 22 fewer questions than the median asker of an original question. Figure 2b shows the distribution of experience heuristic values for original and duplicate questions; visual analysis of Figure 2b supports the observation that 'experienced' users tend to produce fewer duplicates than 'inexperienced' users. A Mann-Whitney U-test indicates that, similar to the reputation experiment, the experience of askers of original questions is significantly larger than that of

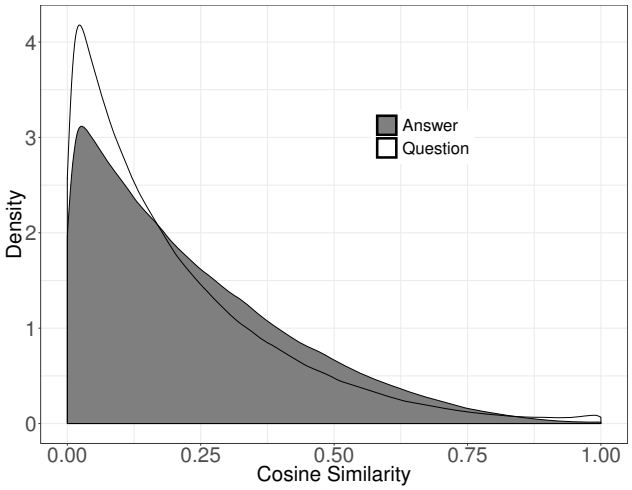


Fig. 3: Duplicate question & answer similarities

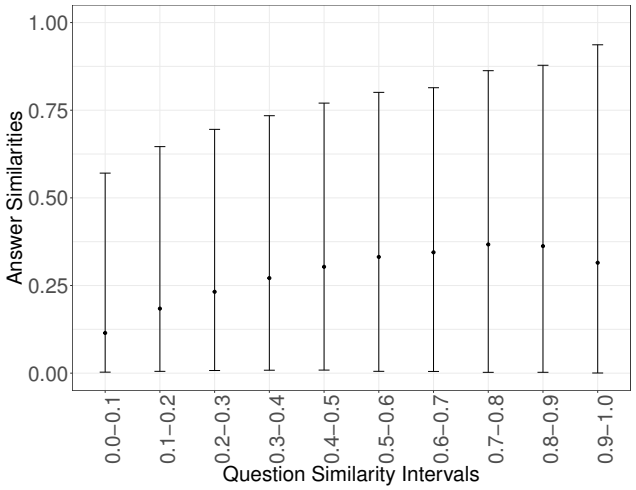


Fig. 4: 95% Confidence interval of answer similarities versus related question similarity

duplicates ($p = 1.4 \times 10^{-4}$). However, unlike the reputation experiment, the Cliff's delta indicates that the practical difference in experience is medium (*delta* $= 0.249$).

> Duplicate questions tend to be posted by users who have not asked many questions previously (regardless of their other activity on Stack Overflow).

RQ2: Do duplicate posts contain unique information?

Approach. Our approach for RQ2 is twofold: (1) we inspect plots of similarity distributions for duplicate questions and answers; and (2) we plot the similarity of duplicate question and answer pairs to analyze their co-relationship.

Observations. Figure 3 shows that duplicate questions and answers exhibit a wide range of similarities including some questions that are completely identical. Indeed, there are many questions and answers that have non-trivial similarities, suggesting that duplicate posts do contain similar information. To ground our analysis against a baseline, we compare the mean cosine similarities of duplicate pair questions (0.204) and answers (0.233) to those of random pairs of questions (0.011) and answers (0.010). The large differences also suggests that the duplicate pairs contain more repeated information than the randomly selected baseline.

Figure 4 shows that as duplicate pair question similarity increase, so too does the similarities of the answers they receive. Visual inspection suggests that as the intervals of question similarities increase, the answer similarity medians plateau and even trend downwards. For example, note the intervals of question similarities between 0.7-0.8 and 0.8-0.9 in Figure 4. In these intervals, the questions are nearing almost perfect similarity, yet the median answer similarities are 0.367 and 0.363 respectively, which suggests that even in the most extreme cases of question similarity, answers still moderately distinct, and can yield alternative (phrasings of) answers.

> Duplicate pairs contain more repeated information than pairs of random questions. Answers of increasingly similar duplicate questions, however, yield relatively distinct answers that could provide value to users.

IV. Conclusions & Future Work

Duplicate posts are the most frequent cause of closure of Stack Overflow posts. While there is a stigma associated with posting duplicate questions, it is reasonable to suspect that duplicates may provide value to the developer community (e.g., alternative phrasings and solutions). This paper presents an initial exploration into the value of duplicate questions and their answers on Stack Overflow. Through investigation into the sources of duplicates and the similarity of duplicate and root questions, we make the following observations:

- Duplicate questions tend to be asked by users who have limited experience posting on Stack Overflow. This does not mean that duplicates are asked by users with little contribution to the Stack Overflow community—user reputation is not correlated with the creation of duplicates.
- Duplicate posts contain varying degrees of duplicate information. Indeed, duplicate questions and answers do contain repeated information, but many answers are relatively unique and could provide additional insight into the topics that they address.

There are promising avenues for future work. First, we plan to explore more precise techniques for computing post similarity (e.g., using word embeddings). Second, we plan to apply sentiment analysis to grapple with another perspective of duplicate question and answer content. Third, since duplicates are not all created equal, we plan to study usefulness of duplicates, aiming to deliver a meaningful quantitative measure that can be used to assess the value of duplicate posts.

REFERENCES

[1] K. Aggarwal, T. Rutgers, F. Timbers, A. Hindle, R. Greiner, and E. Stroulia. Detecting duplicate bug reports with software engineering domain knowledge. In *Proceedings of the International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, pages 211–220, 2015.

[2] M. Ahasanuzzaman, M. Asaduzzaman, C. K. Roy, and K. A. Schneider. Mining duplicate questions in Stack Overflow. In *Proceedings of International Conference on Mining Software Repositories (MSR)*, pages 402–412, 2016.

[3] A. Bacchelli, L. Ponzanelli, and M. Lanza. Harnessing stack overflow for the ide. In *Proceedings of the International Workshop on Recommendation Systems for Software Engineering (RSSE)*, pages 26–30, 2012.

[4] K. Bajaj, K. Pattabiraman, and A. Mesbah. Mining questions asked by web developers. In *Proceedings of the Working Conference on Mining Software Repositories (MSR)*, pages 112–121, 2014.

[5] S. Baltes, C. Treude, and S. Diehl. SOTorrent: Studying the origin, evolution, and usage of Stack Overflow code snippets. In *Proceedings of the International Conference on Mining Software Repositories (MSR)*, 2019.

[6] A. Barua, S. W. Thomas, and A. E. Hassan. What are developers talking about? an analysis of topics and trends in stack overflow. *Empirical Software Engineering (EMSE)*, 19:619–654, 2012.

[7] B. Bazelli, A. Hindle, and E. Stroulia. On the personality traits of stackoverflow users. In *Proceedings of the International Conference on Software Maintenance (ICSM)*, pages 460–463, 2013.

[8] N. Bettenburg, R. Premraj, T. Zimmermann, and S. Kim. Duplicate bug reports considered harmful ... really? In *Proceedings of the International Conference on Software Maintenance (ICSM)*, pages 337–345, 2008.

[9] S. Bird, E. Klein, and E. Loper. *Natural Language Processing with Python*. O'Reilly Media, Inc., 1st edition, 2009.

[10] D. Correa and A. Sureka. Fit or unfit: Analysis and prediction of 'closed questions' on stack overflow. In *Proceedings of the Conference on Online Social Networks (COSN)*, pages 201–212, 2013.

[11] D. Hoogeveen, A. Bennett, Y. Li, K. Verspoor, and T. Baldwin. Detecting misflagged duplicate questions in community question-answering archives. In *Proceedings of the International AAAI Conference on Web and Social Media (ICWSM)*, 2018.

[12] E. Juergens, F. Deissenboeck, B. Hummel, and S. Wagner. Do Code Clones Matter? In *Proceedings of the International Conference on Software Engineering (ICSE)*, pages 485–495, 2009.

[13] D. Luciv, D. V. Koznov, G. Chernishev, A. N. Terekhov, K. Romanovsky, and D. A. Grigoriev. Detecting near duplicates in software documentation. *Programming and Computer Software*, 44(5):335–343, 2018.

[14] A. Pal, R. Farzan, J. A. Konstan, and R. E. Kraut. Early detection of potential experts in question answering communities. In *Proceedings of the International Conference on User Modeling, Adaption, and Personalization (UMAP)*, pages 231–242, 2011.

[15] L. Ponzanelli, G. Bavota, M. Di Penta, R. Oliveto, and M. Lanza. Mining stackoverflow to turn the ide into a self-confident programming prompter. In *Proceedings of the Working Conference on Mining Software Repositories (MSR)*, pages 102–111, 2014.

[16] Y. Zhang, D. Lo, X. Xia, and J.-L. Sun. Multi-factor duplicate question detection in Stack Overflow. *Journal of Computer Science and Technology*, 30(5):981–997, 2015.