A Replication of "An Empirical Study on Developer Interactions in StackOverflow"

JORY ANDERSON, University of Victoria, Canada
CASSANDRA CUPRYK, University of Victoria, Canada
NIMMI WEERADDANA, University of Victoria, Canada
YIMING SUN, University of Victoria, Canada

Additional Key Words and Phrases: Stack Overflow, Topic Modeling, LDA

ACM Reference Format:
Jory Anderson, Cassandra Cupryk, Nimmi Weeraddana, and Yiming Sun. 2020. A Replication of "An Empirical Study on Developer Interactions in StackOverflow". *ACM Trans. Graph.* 1, 1 (December 2020), 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Software developers often seek answers to solve tasks such as bug fixing, adding a new feature, changing an existing feature, etc. and can find these answers on question and answer (Q&A) websites [1]. One of the most popular Q&A websites used by developers is Stack Overflow. Stack Overflow has 14 million users and about 10 million visits to the site per day [2]. Even with this large amount of user traffic, there remains 6,175,850 questions on Stack Overflow that have no upvoted or accepted answers out of the total 20,476,476 questions as of November 26, 2020. [3].

This could be due to a number of reasons. Firstly, Stack Overflow is a company that has a number of competitors: DaniWeb, DreamInCode, Codecall, Bytes, DevShed, and CodingForums [4]. Thus, Stack Overflow may be losing developers who can answer these questions to their competitors. Secondly, the lack of answers could be due to the fact that Stack Overflow is not always considered to be a welcoming site for all users [5]. In fact, Stack Overflow had to create general guidelines on how users should conduct themselves on Stack Overflow, such as "Let's stop judging users for not knowing things" and "Let's make it easier for new users to succeed" [5].

Moreover, the authors of this study decided to take an in-depth look at the distribution of answers and questions posted by users on Stack Overflow. As a result, this study replicated the paper titled "An Empirical Study on Developer Interactions in StackOverflow" by Wang et al., which mined and analyzed data of user activity on Stack Overflow, more specifically mined the questions and answers that users had posted on Stack Overflow [6]. Therefore, the expected results of this study would be similar to the results obtained by Wang et al.

Authors' addresses: Jory Anderson, University of Victoria, , Victoria, British Columbia, Canada; Cassandra Cupryk, University of Victoria, , Victoria, British Columbia, Canada; Nimmi Weeraddana, University of Victoria, , Victoria, British Columbia, Canada; Yiming Sun, University of Victoria, , Victoria, British Columbia, Canada.

0730-0301/2020/12-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1.1 Motivation

The motivation behind replicating Wang et al.'s study would be to benefit Stack Overflow, since the results could better inform Stack Overflow about the general activity of the users on their website. In addition, Stack Overflow's revenue depends on a high amount of user activity on their website, since a large portion of their business model depends on companies paying Stack Overflow to host advertisements as well as to connect users with job offers [7]. Moreover, the more users interact with Stack Overflow, the more revenue Stack Overflow would earn.

1.2 Research Questions

In this study, the following research questions were answered:

- RQ1: What is the distribution of the amount of questions that each user posts?
- RQ2: What is the distribution of the amount of answers that each user posts?
- RQ3: Do users that ask questions answer questions too?
- RQ4: What topics do users ask about and what is the distribution of the topics?

1.3 Organization of the report

The next section discusses the background and related work of this study. The methodology and results are presented in Sections 3 and 4, respectively. The results are then discussed in Section 5, followed by the limitations, future work and finally concluding the study.

2 BACKGROUND AND RELATED WORK

This section provides definitions on terminologies referred to in the rest of this report, examples of various methodologies categorizing Stack Overflow questions and answers, and knowledge that was borrowed from related works.

2.1 Terminology

The rest of this study will refer to the following terminologies:

- *Post*: Each question or answer is referred to as a post.
- *Users*: The authors of the sampled questions and the authors of the questions' associated answers.

2.2 Categorization of Stack Overflow

Categorizing Stack Overflow posts have already been done in a number of studies. Beyer et al. manually categorized 450 Android-related Stack Overflow posts concerning their question and problem types[8]. In a later study, Beyer et al. harmonized the existing categorizations of Stack Overflow into a single categorization [9]. In addition, Nasehi et al. conducted a grounded theory study to categorize the 163 Stack Overflow questions, which had the *java* tag [1]. Treude et al. utilized the title text and the body text in order to categorize questions into different question types (ex. how-to, review, error, etc.) [10]. In addition, Treude et al. manually categorized 385 Stack Overflow questions using the question tags in order to discover the Computer Science topics that were discussed in Stack Overflow (ex. Programming language, framework, environment, etc) [10].

In contrast to Treude et al. [10], Wang et al. [6] categorized Stack Overflow questions using an automated approach, i.e., topic modeling, which is a powerful technique for text mining and finding relationships among text documents [11]. Moreover, Wang et al. used Latent Dirichlet allocation (LDA) which is a popular topic modeling tool among the several topic modeling tools. LDA has also been used in a number of studies to analyze Stack Overflow posts. Allamanis et al. identified question types and topics of Stack Overflow questions using LDA [12]. Next, Rosen et al. utilized an LDA in order to figure out which challenges mobile app development users face [13]. In addition, Barua et al. utilized LDA in order to discover the main topics discussed in Stack Overflow questions. Their results demonstrated that the most popular topics are: web development, mobile applications, git and MySQL [14]. In another study, Openja et al. determined 38 release engineering topics by grouping the Stack Overflow questions together using an LDA and then manually assigning a label to each group [15].

2.3 Overview of This Study

In this study, the topic modeling tool, in particular LDA was used to group Stack Overflow questions and these groupings were used to determine the topics. This is due to the fact that topic modeling automatically creates a list of related tokens. These related tokens were then used to determine each topic. Moreover, LDA makes the process of categorizing the questions into topics more efficient in comparison to manual categorization.

As mentioned in the introduction, this study is based on Wang et al.'s study [6]. Wang et al. utilized LDA to determine topics discussed in Stack Overflow. In addition, Wang et al. calculated statistics regarding user participation such as the the distribution of the amount of posts that each user posts.

As background for this study, the research context was determined to be *Industrial* according to Lenberg et al. [16]. In addition, the answers for the Who-What-How framework [17] were determined to further verify that Wang et al.'s work aligns with the motivation of this study and the answers are listed below.

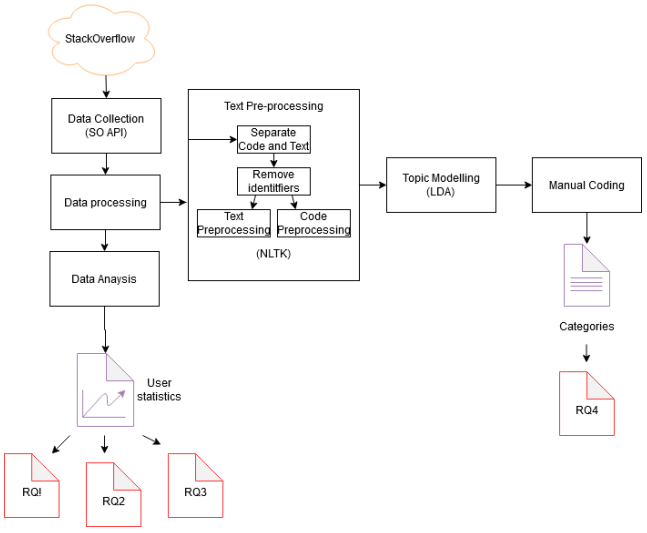
(1) *Who does this benefit?* Human Stakeholder - Organization, in particular, Stack Overflow
(2) *What is the main research contribution type?* Descriptive Analysis
(3) *How is the research conducted?* Data Strategy, in particular using Stack Overflow data

Accordingly, Wang et al.'s study [6] acted as a guide for all the work done in this study. However, this study made the following modifications to Wang et al.'s work. Firstly, the research questions 1, 2 and 4 slightly deviated from Wang et al. Secondly, the data collection of this study was based on reverse chronological order whereas Wang et al.'s work was based on random sampling. In addition, the text preprocessing was improved by replacing the HTML entities with a space character. Next, the code preprocessing was improved to tokenize *camelCase* identifiers. For the LDA, this study utilized python sklearn library [18] whereas the Wang et al. utilized the Java implementation *JGibbLDA* [19]. Then, the approaches defined in Wang et al. were modified in order to determine which questions belonged to each topic. Finally, in contrast to Wang et al., the authors of this study did not create their own unique list of topics, rather the topic list generated by Openja et al. [15] was used to manually assign labels to the unlabeled topics.

3 METHODOLOGY

Stack Overflow questions, answers, and public user profile information are retrievable from the Stack Exchange API [20]. For the first three research questions, statistics regarding user participation are generated after mapping each user to their questions and answers. To answer the fourth research question, the title, text, and answers for each question were processed further using text processing tools. The LDA was then used to capture the words that would be necessary for determining the topics. Figure 1 presents an overview of the methodology, and the following subsections provide the specific details of the methodology.

Fig. 1. Methodology



3.1 Data Collection

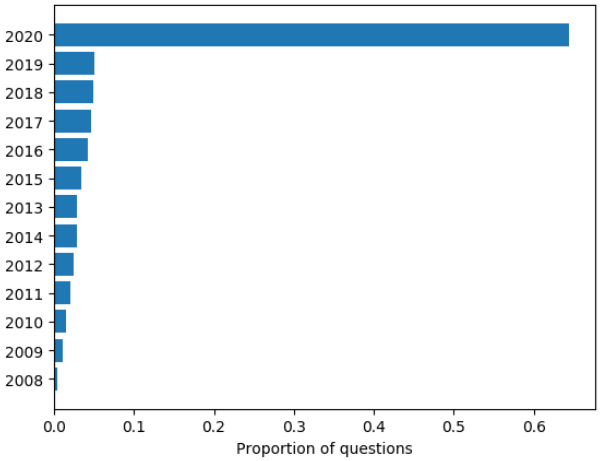
Using the Stack Exchange API's default sorting algorithm, the API would return the 100 most recently active questions for each API

call [21]. In addition, the questions were sampled in reverse chronological order from October 31^{st}, 2020 as a means of capturing the current topics being discussed on Stack Overflow and the current user activity on Stack Overflow.

The authors of this study collected the dataset during the period of November 14^{th}, 2020 to November 21^{st}, 2020. Each question or answer contained the post ID, the text pertaining to that question or answer, and the user ID of the author. For every question, the set of answers was also collected. In addition, the collected data was formatted in HTML. Thus, ~236,000 questions were collected from the API, and after removing duplicates there were more than 100,000 unique questions. Duplicate questions were identified by finding posts with the same post ID. Finally, the sample was rounded down to exactly 100,000 distinct questions, and the number of distinct users who post questions and/or answers is 209,302.

In the dataset of this study, 100,000 questions were asked by 84,386 users, and answered by 142,665 users (ratio 1:1.69). While in Wang et al.'s study, 63,863 questions were asked by 44,087 users, and answered by 43,055 users (ratio 1:0.98). In addition, Figure 2 shows the distribution of years when the questions were created. The overall time frame of the questions is 13 years, yet most of the questions were created in the year of 2020. Also, there are 20,191 distinct tags in the 100,000 sampled questions. Figure 3 shows the 20 most frequent tags.

Fig. 2. Creation years of questions

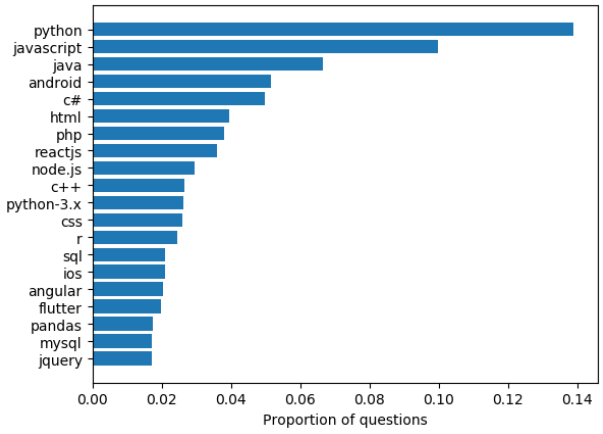


3.2 Data Preprocessing

Firstly, the collected HTML documents were separated into text and code. Secondly, the text and the code were preprocessed separately as described below.

3.2.1 ***Separating Text and Code***. Similar to the original paper [6], the content of each post is split into a text part and a code part. Since the content in the dataset was formatted in HTML, the code part was considered to be the block(s) tagged *<code>*, and the rest was considered to be the text part.

Fig. 3. Top 20 Frequent Tags



3.2.2 ***Text Processing***. For the text preprocessing step, the Natural Language Toolkit (NLTK) [22] was used on the texts and the general outline of preprocessing was followed, i.e., tokenization, normalization, and noise removal [23]. For the tokenization step, the texts were tokenized into words. Each non-alpha numeric character, and each HTML entity was replaced with a space character and tokenized, which is an improvement of the original paper. The list of HTML entities were obtained from [24]. At the normalization step, the words were converted to lowercase, lemmatized, and the stop words were removed. All the tokens that had a length of less than 3 characters were also removed in order to further remove any noise [25].

3.2.3 ***Code Processing***. For the code preprocessing step, keywords listed in [26] and non-alpha numeric characters were removed from the codes in order to retain only the identifier and the comments. The identifiers and the comments in the codes were subjected to tokenization, stemming and stop word removal in the same way as for the text. Also, the original methodology was improved by splitting the camel case into sensible tokens [27]. In addition, the users often use *<code>* blocks for presenting REPL commands, outputs and error messages. However, this was still considered as code and the same preprocessing was applied.

3.3 RQ1 & RQ2: The distribution of the number of posts

To answer the first two research questions, the distributions of the amount of questions or answers each user posted was captured.

3.4 RQ3: Do users that ask questions answer questions too?

To answer RQ3, a similar approach to the original study [6] was taken. The questions and answers were collected from each user who asked a question in the dataset. Then, the proportion of answers in relation to the user's total number of posts was calculated by the following formula: 1.

$$answer\_proportion = \frac{answer\_count}{question\_count + answer\_count} \quad (1)$$

3.5 RQ4: What topics do users ask about and what is the distribution of the topics?

To answer RQ4, topic modeling, in particular, LDA was applied to the preprocessed dataset, and was followed by a manual coding session. The methodology is explained in the following subsections.

3.5.1 ***Topic Modeling.*** In order to achieve topic modeling, the question and the associated answers for that particular question were considered to be a document. Moreover, a document included the title, the question body, the codes in the question body, and all the associated answers including the answer text and any answer codes which were already preprocessed.

In this study, LDA was used, which is the the same topic modeling technique used in Wang et al.'s study [6]. For the implementation of LDA, the python package Sklearn was used [18]. 5 topics were obtained as done in [6]. In addition, the chosen number of tokens associated with each unlabeled topic was 15.

Next, three approaches were utilized in order to determine the number of questions that belonged to each unlabeled topic. Using the LDA and the unlabeled topic's associated tokens, the probability of each question being assigned to the unlabeled topic was calculated. All three approaches used these probabilities in order to assign the questions to the correct unlabeled topic. The approaches differed in the way of counting the questions for each unlabeled topic. The first approach is the same as the first approach of the Wang et al.'s study [6]. The second approach is a modification to the second approach of the Wang et al.'s study. The final approach was defined by the authors of this study.

Approach 1: Top-1 Topic Assignment Recall that the probability of each question being assigned to an unlabeled topic was calculated. Thus, the highest probability would determine which unlabeled topic the question would be assigned to. Once the question was assigned to that unlabeled topic, the counter of number of questions for that unlabeled topic would be incremented by 1.

Approach 2: Weighted Topic Assignment In this approach, a question could be assigned to multiple unlabeled topics. If a question had a probability of belonging to an unlabeled topic t that is equal to 0.2 or higher, but less than 0.5, the counter for the unlabeled topic t was incremented by 0.2. If a question had a probability of belonging to an unlabeled topic t that was greater than or equal to 0.5, the counter for the unlabeled topic t was incremented by 1. However, if a question had a probability of belonging to an unlabeled topic t that is less than 0.2, the counter for the unlabeled topic t was not incremented at all. Thus, for approach 2, the only possible increments for the unlabeled topics' counters were constant values, i.e., 0, 0.2 and 1.

Approach 3: Fuzzy Topic Assignment A question could be assigned to multiple unlabeled topics as in approach 2, but when incrementing the counter for the unlabeled topics, the counter was incremented by the probability rather than constant values, such as 0, 0.2, 1. Therefore, the authors of this study named this approach as *the fuzzy topic assignment approach.*

3.5.2 ***Manual Coding.*** The 18 most frequently occurring topics were obtained from [15] including Software Testing, Build Failure, Branching Remote Upstream, Merge Conflict, Ansible, Msbuild, Configuration Management, Security, Mobile Deployment, Continuous Deployment, Web Deployment, etc. Some of these topics were merged based on the domain of the topic. For example, the topics Branching & Remote Upstream and Merge Conflict merged into a single topic named *Version Controlling*. Similarly, Ansible which is a configuration management tool was merged to *Configuration Management*. The updated list of topics were Software Testing Frameworks & Code Reviews, Software Building, Version Controlling, Configuration Management, Security, Mobile Development, Continuous Integration & Deployment, Web Development and Testing, Virtualization, Rollback, and File Tranforms.

The manual coding process was conducted as follows. One of the authors acted as a moderator. The moderator randomly sampled five questions from each of the unlabeled topics detected by LDA. The three other authors manually and independently coded the questions in order to assign a label to the unlabeled topics. The label was selected from the list of topics derived from [15]. Finally, the discrepancies of the topic labels chosen by the authors were then discussed until the authors reached an agreement on the topic label.

4 RESULTS

The following sections present the results for each research question.

4.1 RQ1: What is the distribution of the amount of questions that each user posts?

Graph (a) in Figure 4 illustrates the number of questions asked by a user in relation to the number of users (the y-axis is in logarithmic scale). Graph (b) in Figure 4 illustrates the number of questions asked by a user (the x-axis is in logarithmic scale) in relation to the accumulated proportion of users.

In the dataset, ~88% (74,604 users) of the users only asked 1 question. As the number of questions posted by a user increased, the number of users decreased exponentially. The maximum number of questions asked by a user was 35 (only 1 user). In addition, the proportion of users who asked more than 5 questions was ~0.4% of users (280 users).

4.2 RQ2: What is the distribution of the amount of answers that each user posts?

Graph (a) in Figure 5 illustrates the number of answers posted by a user in relation to the number of users (the y-axis is in logarithmic scale and the x-axis is capped to 100). As the number of answers posted by a user increased, the number of users decreased exponentially. Graph (b) in Figure 5 illustrates the number of answers posted by a user (the x-axis is in logarithmic scale) in relation to the accumulated proportion of users. The maximum number of answers posted by a user was 343. Specifically, ~79% (112,996 users) of the users only posted 1 answer, while only 2.4% (3,419 users) of the users posted more than 5 answers.

Fig. 4. Distribution of Questions

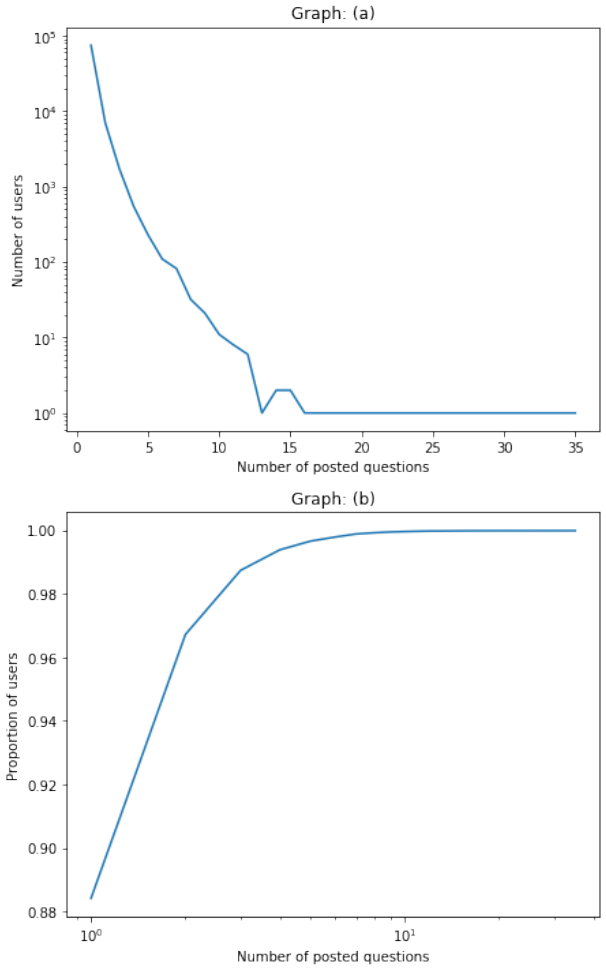
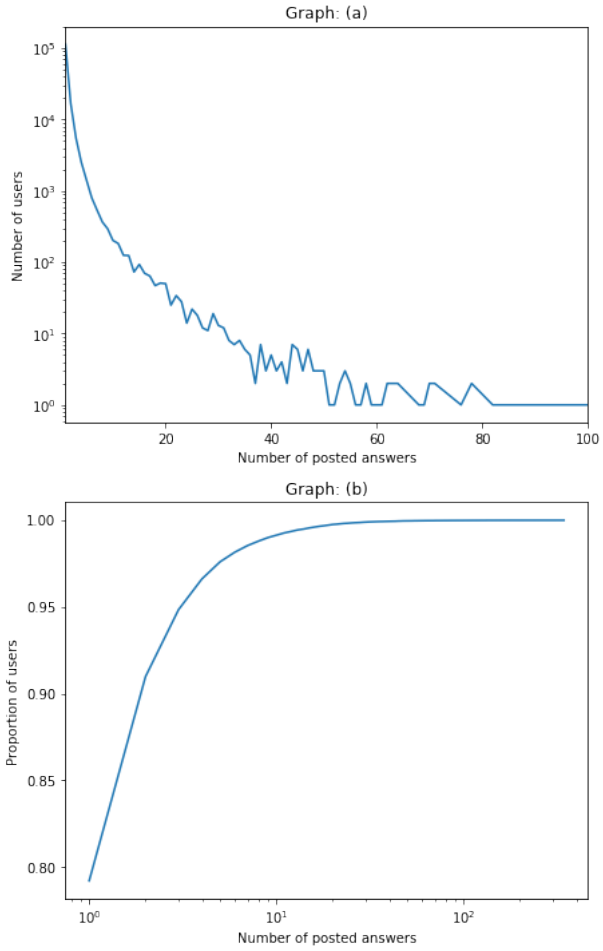


Fig. 5. Distribution of Answers

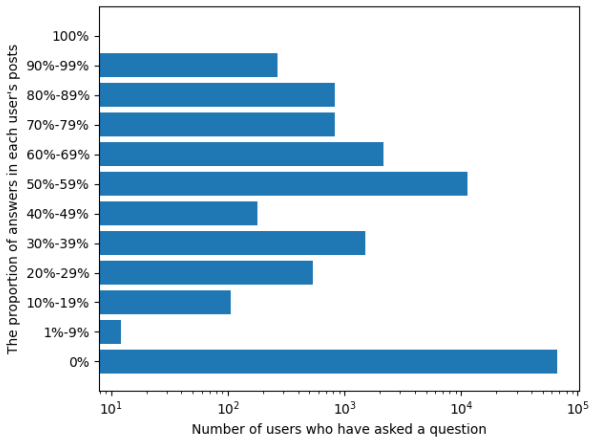


4.3 RQ3: Do users that ask questions answer questions too?

The next step was to find the number of questions and answers each user has posted, from those who had asked a question in our sample. After doing so, for each user their answer proportion was determined (see equation 1) and plotted in Figure 6 (logarithmic x-axis).

Figure 6 demonstrates there is a large proportion of users from the sampled questions that have not posted any answers. That is, ~79% (66,637 users) of users who have asked at least one question do not attempt to answer other questions. The remaining ~21% (17,749 users) of users in the sample have submitted at least one or more answers to Stack Overflow. Of the users who have posted answers, ~18% (15,412 users) of users have posted more answers than questions, and ~82% (68,974 users) of users have posted more questions than answers.

Fig. 6. RQ3: Do users that ask questions answer questions too?



4.4 RQ4: What topics do users ask about and what is the distribution of the topics?

This research question was answered using topic modeling as mentioned in Section 3. As LDA generates a list of tokens representing a topic, the authors in this study manually labeled the topics as mentioned in Section 3.

Table 1 shows the list of related tokens per topic, the manually labeled topics, and the number of questions that belong to each topic. Figure 7, Figure 8, and Figure 9 shows the proportions of the questions belonging to each topic according to approach 1, approach 2, and approach 3 defined in Section 3.

Figure 10 presents for each topic, an overview of the number of unique users who posted questions, the number of unique users who posted answers, and the number of unique users. Approach 1 (Top-1) was used to count the unique users and a unique user was defined as a user who had asked multiple questions, but was counted as one user.

Fig. 7. Topic Distribution (Approach 1: Top-1)

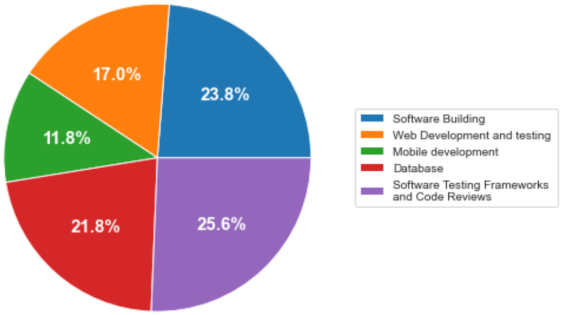
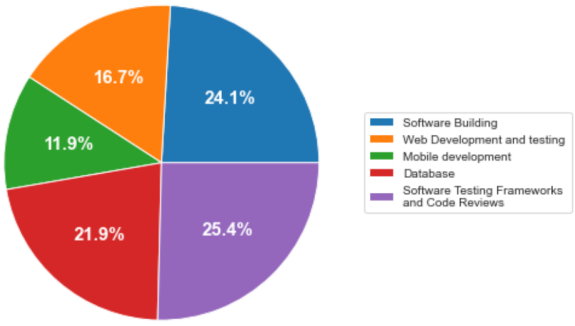


Fig. 8. Topic Distribution (Approach 2: Weighted)



5 DISCUSSION

As mentioned previously, the questions were sampled reverse chronologically from late October instead of randomly sampling. Thus, this explains why Figure 2 presents a predominant number of questions from 2020. In addition, the default sorting for questions returned

Fig. 9. Topic Distribution (Approach 3: Fuzzy assign)

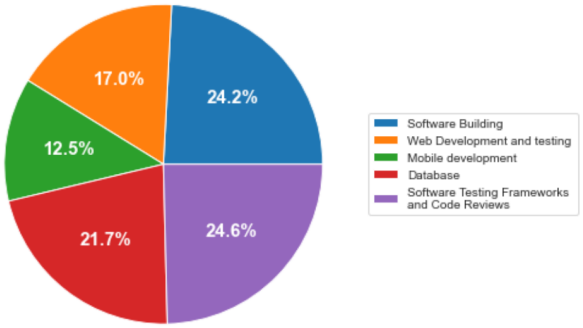
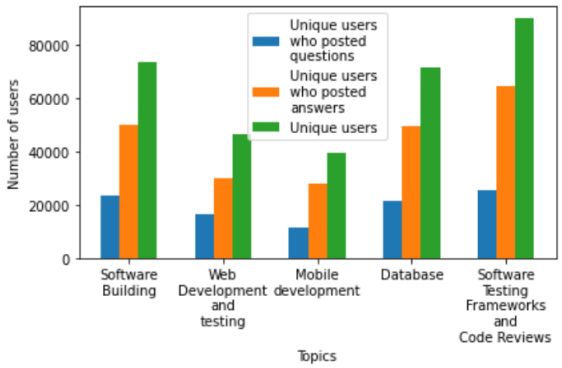


Fig. 10. Users contributed per topic (Approach 1: Top-1)



from the Stack Exchange API is set to the setting of 'activity' [20]. This setting would cause the API to capture questions asked this year as well as older questions still receiving traffic and revisions to date. Thus, this sorting algorithm captures more recently discussed topics.

5.1 RQ1, RQ2, and RQ3

To answer RQ1, recall that Figure 4 showed that the number of posted questions by a user decreased as the number of users increased. In addition, the RQ1 results found that ~88% (74,604 users) of the users had only asked 1 question and 0.4% of users had asked more than 5 questions. In comparison, Wang et al.'s study found that 77% of the users had only asked 1 question, and 1.6% had asked more than 5 questions.

Similarly, to answer RQ2, Figure 5 showed that the number of posted answers by a user decreased as the number of users increased. In addition, the RQ2 results found that ~79% (112,996 users) of the users only posted 1 answer, while only 2.4% (3,419 users) of the users posted more than 5 answers. In comparison, Wang et al.'s study found that 66% of the users had only posted 1 answer, and 8% of the users had posted more than 5 answers.

To answer RQ3, the majority of users do not answer questions at all. Only ~21% of users answered a question within the sample and only ~18% of users answered more questions than asked questions.

Table 1. Related tokens, manually labeled topic, and the number of questions

	Related Tokens	Manually Labeled Topic	Number of Questions
0	work nan one call class need object get system function method like type code use	Software Building	23757
1	email queri get product post user put url http code script request html page use	Web Development and testing	16963
2	ctent com size background app text imag lay use development item view style height wth col	Mobile development	11841
3	want data function array like key put column size code result time valu number use	Database	21795
4	instal tri nan error server project work build path com app run http use file com app run http use file	Software Testing Frameworks and Code Reviews	25644

In terms of shape of the graph, the results presented in Figure 6 resemble that of the Wang et al.'s results for RQ3 [6]. In comparison to Wang et al.'s results [6], the same increase in proportions among users at 50%-59% exists, and similar absences of proportions exist among users in the 1%-9%, 40%-49%, and 100% ranges. These results indicate that even though more users are answering questions since 2013, the overall proportion of active users has not changed from 2013 when the study was originally conducted [6].

Next, there could be a number of barriers that prevent users from participating on Stack Overflow [28]. Denae et al. highlighted 3 main topics titled: *the Muddy Lens Perspective*, *the Impersonal Interactions*, and *On-Ramp Roadblocks*. *The Muddy Lens Perspective* highlights the fact that people do not comprehend how Stack Overflow works. *The Impersonal Interactions* indicated how people feel that it's difficult to form personal connections on Stack Overflow. Thus, users may take some time to learn about their community and then proceed to post at a later date once they feel comfortable. Other users want to avoid negative interactions that they've seen occur within the community. The *On-Ramp Roadblocks* highlighted the obstacles that prevent users from being interested in posting on Stack Overflow, such as time constraints on interacting with Stack Overflow [28]. Finally, the most popular reason is that users were able to find the answer to the question without engaging with the community [28].

More specifically, the lack of posted answers on Stack Overflow could be explained by a few reasons. Firstly, the lack of answers for a question could be due to the quality of the question. For example, questions written in a neutral emotional style, questions providing sample code and data, and questions using capital letters when it's appropriate have higher chances of getting answered. Next, the timing of posting a question can affect the probability of receiving an answer [29]. For example, a user has a higher probability of receiving an answer on the weekend [30]. In addition, Avigit et al. found there is a higher probability that a question will get an accepted answer if the question contains a higher number of tags [31].

5.2 RQ4: What topics do users ask about and what is the distribution of the topics?

In Wang et al.[6]'s study, the following topics were identified: *User Interface*, *Stack Trace*, *Large Code Snippet*, *Web Document* and *Miscellaneous*. The obtained topics of this study are different from the topics in Wang et al.'s study. The main difference is that the Wang et al.'s results determined the *Miscellaneous* topic had the highest number of questions, while the *Miscellaneous* topic was not identified based on this study's methodology.

Next, this study found that for all three approaches the topic called *Software Testing Frameworks and Code Reviews* had the highest proportion of questions as shown in Fig. 7, Fig. 8, and Fig. 9. This finding is supported by Openja's work, since Openja's study found that *Software Testing* was the most commonly observed release engineering topic on Stack Overflow [15]. In addition, Wang et al. found that *Web Document* had the second highest number of questions [6]. However, the proportion of questions belonging to *Web Development and Testing* for this study was not the second highest, but rather the fourth highest. Finally, the majority of the users asked questions and/or answers related to the topic titled *Software Testing Frameworks and Code Reviews*.

6 LIMITATIONS

Overall, the following are the potential limitations to this study:

- Recall that the topics were determined using the manual coding assignment done by the authors of this study. Thus, this could have introduced some experimenter bias. In addition, only 5 questions were used to determine each topic. Moreover, a larger number of questions should have been considered in order to determine the topics and this would have increased the precision of the results.
- The ambiguity in the words was not taken into consideration in either the original work or this study. For example, *post* could either be referring to a Stack Overflow post or *POST request*, which could have affected the categorization of the topics.
- A potential limitation for RQ3 is that the sample size of the dataset may have been too small. For example, a user may

have asked a question that was included in the dataset and answered a question that was not included in the dataset. Moreover, increasing the sample size could have reduced this issue.

- In addition, another potential limitation for RQ3 was collecting the answers and questions at separate times. This was an issue, since there happened to be a few occasions where the questions had been collected, but then had been deleted off the Stack Overflow website. Thus, a user could have asked a question and answered another question. However, between the time that the questions had been collected and the answers had been collected, the question that the user had answered could have been deleted. Thus, the fact that the user had asked a question and had also answered a question had not been recorded and would have affected the RQ3 results.

7 FUTURE WORK

To begin, a future study would have answered RQ4 from the Wang et al.'s study [6]. In addition, this study would include a larger dataset of questions and answers in order to obtain more generalizable results. Next, the limitations section highlighted that 5 questions were considered in order code each topic. Ideally, future research would look to increase the number of questions sampled when labeling topics. In addition, using a wider set of topics would introduce more realistic results.

Next, there are some interesting avenues for future work. Firstly, there was an official Stack Overflow survey conducted in 2020. Thus, it would have been interesting to compare the qualitative data obtained by Stack Overflow with the quantitative results of this study. Next, the popular topics within each programming language could be identified to provide more insight on the discussions occurring on Stack Overflow. In addition, due to time constraints, the list of topics that were considered to categorize the questions were drawn solely from Openja et al. [15]. Moreover, in order to improve the realism of the study, a larger amount of topics to categorize the Stack Overflow questions should be obtained.

Another study potentially worth investigating is how can the number of active users on Stack Overflow be increased. Thus, a qualitative study could be done in order to find out the reasons why users do or do not post on Stack Overflow. In addition, further work could look into the turn-around time of solved questions with particular scores and figure out why questions do or do not have quick turn-around time [29]. This could even be extended further by discovering what users on Stack Overflow consider to be a good post [29].

8 CONCLUSION

In this study, the authors replicate Wang et al.'s paper "An Empirical Study on Developer Interactions in Stack Overflow", aiming to help Stack Overflow enhance their user engagement by exploring the general activity of the users. Moreover, the topic modeling, in particular the LDA was utilized to determine the topics of the Stack Overflow questions. In addition, the following modifications were performed to the Wang et al.'s study, such as deviating research questions, using an alternative sampling algorithm, applying tokenization to code identifiers, modifying weighted topic assignment, introducing fuzzy topic assignment approach, and labeling topics manually from a given list.

The results show that, from the sample: the number of users who post questions is about half of who post answers (74,604 vs. 142,665); dominant proportions of users (99.6%/97.6%) only post a very few numbers of questions/answers (less than 6); 79% of the users who post questions do not contribute Stack Overflow back by posting answers. Finally, from topic modeling, there is a consistency in topic distributions among the results from all the three approaches, and the top 3 topics are Software Testing Code Reviews, Software Building, and Database.

REFERENCES

[1] S. M. Nasehi, J. Sillito, F. Maurer, and C. Burns, "What makes a good code example?: A study of programming q&a in stackoverflow," in *2012 28th IEEE International Conference on Software Maintenance (ICSM)*, pp. 25–34, IEEE, 2012.

[2] "Stackexchange questions per day." https://stackexchange.com/sites?view=list#questionsperday. Accessed 2020-12-10.

[3] "All questions," Nov 2020.

[4] J. Geek, "What are the main competitor sites to stack overflow?," Oct 2019.

[5] J. Hanlon, "Stack overflow isn't very welcoming. it's time for that to change.," Apr 2018.

[6] S. Wang, D. Lo, and J. Lingxiao, "An empirical study on developer interactions in stackoverflow," *Symposium On Applied Computing*, March 2013.

[7] N. Craver, "How we make money at stack overflow: 2016 edition," Nov 2016.

[8] S. Beyer and M. Pinzger, "A manual categorization of android app development issues on stack overflow," in *2014 IEEE International Conference on Software Maintenance and Evolution*, pp. 531–535, IEEE, 2014.

[9] S. Beyer, C. Macho, M. Di Penta, and M. Pinzger, "Automatically classifying posts into question categories on stack overflow," in *2018 IEEE/ACM 26th International Conference on Program Comprehension (ICPC)*, pp. 211–21110, IEEE, 2018.

[10] C. Treude, O. Barzilay, and M.-A. Storey, "How do programmers ask and answer questions on the web? (nier track)," *ICSE '11: Proceedings of the 33rd International Conference on Software Engineering*, May 2011.

[11] H. Jelodar, Y. Wang, C. Yuan, X. Feng, X. Jiang, Y. Li, and L. Zhao, "Latent dirichlet allocation (lda) and topic modeling: models, applications, a survey," *Multimedia Tools and Applications*, vol. 78, no. 11, pp. 15169–15211, 2019.

[12] M. Allamanis and C. Sutton, "Why, when, and what: analyzing stack overflow questions by topic, type, and code," in *2013 10th Working Conference on Mining Software Repositories (MSR)*, pp. 53–56, IEEE, 2013.

[13] C. Rosen and E. Shihab, "What are mobile developers asking about? a large scale study using stack overflow," *Empirical Software Engineering*, April 2015.

[14] A. Barua, S. Thomas, and A. Hassan, "What are developers talking about? an analysis of topics and trends in stack overflow.," *Empirical Software Engineering*, November 2012.

[15] M. Openja, B. Adams, and F. Khomh, "Analysis of modern release engineering topics:–a large-scale study using stackoverflow–," in *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pp. 104–114, IEEE, 2020.

[16] P. Lenberg, R. Feldt, and L. G. Wallgren, "Behavioral software engineering: A definition and systematic literature review," *Journal of Systems and software*, vol. 107, pp. 15–37, 2015.

[17] M.-A. Storey, N. A. Ernst, C. Williams, and E. Kalliamvakou, "The who, what, how of software engineering research: a socio-technical framework," *Empirical Software Engineering*, vol. 25, no. 5, pp. 4097–4129, 2020.

[18] "Topic modeling in python: Latent dirichlet allocation (lda)." https://towardsdatascience.com/end-to-end-topic-modeling-in-python-latent-dirichlet-allocation-lda-35ce4ed6b3e0. Accessed: 2020-11-27.

[19] "Jgibblda." http://jgibblda.sourceforge.net/. Accessed: 2020-12-05.

[20] "Stack exchange api." https://api.stackexchange.com. Accessed 2020-12-06.

[21] "Python wrapper for stackapi." https://pypi.org/project/StackAPI/. Accessed: 2020-12-06.

[22] "Natural language toolkit." https://www.nltk.org/. Accessed: 2020-12-06.

[23] "Nlp text preprocessing: A practical guide and template." https://towardsdatascience.com/nlp-text-preprocessing-a-practical-guide-and-template-d80874676e79. Accessed: 2020-11-25.

[24] "Html entities." https://www.w3schools.com/html/html_entities.asp. Accessed: 2020-11-27.

[25] "Text preprocessing: Removal of punctuations." https://studymachinelearning.com/text-preprocessing-removal-of-punctuations/. Accessed: 2020-12-05.

[26] "Reserved key words list of various programming languages." https://github.com/AnanthaRajuCprojects/Reserved-Key-Words-list-of-various-programming-languages. Accessed: 2020-11-25.

[27] "How to do camelcase split in python." https://stackoverflow.com/questions/29916065/how-to-do-camelcase-split-in-python. Accessed: 2020-11-27.

[28] D. Ford, J. Smith, P. J. Guo, and C. Parnin, "Paradise unplugged: identifying barriers for female participation on stack overflow," *FSE 2016: Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, November 2016.

[29] F. Calefato, F. Lanubile, and N. Novielli, "How to ask for technical help? evidence-based guidelines for writing questions on stack overflow," *Information and Software Technology*, vol. 94, pp. 186–207, February 2018.

[30] A. Bosu, C. S. Corley, D. Heaton, D. Chatterji, J. C. Carver, and N. K. Kraft, "Building reputation in stackoverflow: An empirical investigation," *2013 10th Working Conference on Mining Software Repositories (MSR)*, pp. 186–207, May 2013.

[31] S. K. Avigit, S. K. Ripon, and K. A. Schneider, "A discriminative model approach for suggesting tags automatically for stack overflow questions," *2013 10th Working Conference on Mining Software Repositories (MSR)*, pp. 73–76, 2013.