Broken External Links on Stack Overflow

Jiakun Liu, Xin Xia, David Lo, Haoxiang Zhang, Ying Zou, Ahmed E. Hassan,and Shanping Li

Abstract—Stack Overflow hosts valuable programming-related knowledge with 11,926,354 links that reference to the third-party websites. The links that reference to the resources hosted outside the Stack Overflow websites extend the Stack Overflow knowledge base substantially. However, with the rapid development of programming-related knowledge, many resources hosted on the Internet are not available anymore. Based on our analysis of the Stack Overflow data that was released on Jun. 2, 2019, 14.2% of the links on Stack Overflow are broken links. The broken links on Stack Overflow can obstruct viewers from obtaining desired programming-related knowledge, and potentially damage the reputation of the Stack Overflow as viewers might regard the posts with broken links as obsolete. In this paper, we characterize the broken links on Stack Overflow. 65% of the broken links in our sampled questions are used to show examples, e.g., code examples. 70% of the broken links in our sampled answers are used to provide supporting information, e.g., explaining a certain concept and describing a step to solve a problem. Only 1.67% of the posts with broken links are highlighted as such by viewers in the posts' comments. Only 5.8% of the posts with broken links removed the broken links. Viewers cannot fully rely on the vote scores to detect broken links, as broken links are common across posts with different vote scores. The websites that host resources that can be maintained by their users are referenced by broken links the most on Stack Overflow – a prominent example of such websites is GitHub. The posts and comments related to the web technologies, i.e., JavaScript, HTML, CSS, and jQuery, are associated with more broken links. Based on our findings, we shed lights for future directions and provide recommendations for practitioners and researchers.

Index Terms—Empirical Software Engineering, Stack Overflow, Broken Link

1 Introduction

Stack Overflow is a valuable knowledge base that serves millions of users around the world [1], [2]. When developers communicate on Stack Overflow, they can use links to introduce the resources that are scattered across the Internet [3], [4]. Based on the Stack Overflow data dump (released on Jun. 2, 2019), among 19,200,512 posts (i.e., questions and answers) and comments, 11,926,354 distinct links are referenced 27,553,546 times in total.

However, with the rapid development of programming-related knowledge, many resources hosted on the Internet are not available anymore. In this paper, we refer to the links that reference to unavailable resources as **broken links**. We also refer to the posts without broken links as **normal posts**, and those with broken links as **broken posts**. In a prior study, Zhang et al. focused on analyzing obsolete knowledge on Stack Overflow and they observed that 11% links in answers are broken links [5]. However, they did not analyze all the broken links on Stack Overflow. Considering a large number of external resources that are referenced by Stack Overflow, it is unclear how Stack Overflow suffers from the broken link problems. Figure 1 shows an example of the comments to the accepted answer that solves the authentication error for a Mifare card[1]. Six comments that were received from Jan 08, 2014, to Feb 17, 2020, complain about the broken link. Broken links can obstruct viewers from getting desired programming-related crowdsourced knowledge on Stack Overflow, and potentially damage the reputation of the Stack Overflow as viewers might regard the broken posts as obsolete [5]. Therefore, it is important to investigate the broken links on Stack Overflow and understand their impacts and characteristics. By doing so, we could provide insights for practitioners and researchers to address this issue.

In our paper, we investigate the broken links on Stack Overflow. To do so, we test the HTTP response status code (i.e., response code) of the 12,446,901 links in all versions of Stack Overflow posts. To mitigate the intermittent behavior, we perform one test using a server located at Virginia, U.S. in Dec 2019, and another test using a server located at Singapore in Jan 2020. We identify the links that are not responded with 2xx (e.g., 200, 201, 202) response code in both trials as broken links. To understand the importance of investigating the broken links on Stack Overflow, we perform a series of preliminary studies on the broken links. We observe that 14.2% of the links are broken links. 404 response code is the most common response code for broken links on Stack Overflow. Links that were posted earlier are more likely to be broken. 22.9% of the links that were in-

- *Jiakun Liu and Shanping Li are with the College of Computer Science and Technology, Zhejiang University, Hangzhou, China. E-mail: {jkliu, shan}@zju.edu.cn*
- *Xin Xia is with the Software Engineering Application Technology Lab, Huawei, Hangzhou, China E-mail: xin.xia@acm.org*
- *David Lo is with the School of Information Systems, Singapore Management University, Singapore. E-mail: davidlo@smu.edu.sg*
- *Haoxiang Zhang is with the Centre for Software Excellence at Huawei, Canada. This work is not related to his role at Huawei. E-mail: haoxiang.zhang@huawei.com*
- *Ahmed E. Hassan is with the Software Analysis and Intelligence Lab (SAIL), Queen's University, Kingston, Ontario, Canada. E-mail: ahmed@cs.queensu.ca*
- *Ying Zou is with the Department of Electrical and Computer Engineering, Queen's University, Kingston, Ontario, Canada. E-mail: ying.zou@queensu.ca*
- *Xin Xia is the corresponding author.*

1. https://stackoverflow.com/q/15881962/

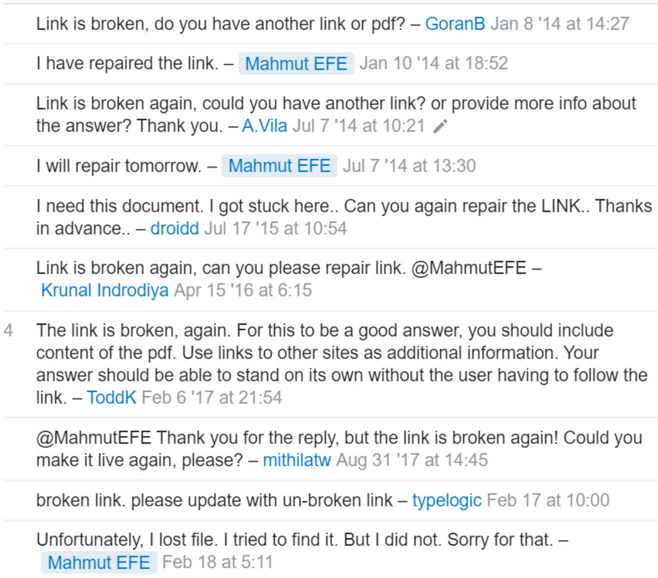


Fig. 1: An example of the comments to the accepted answer with a broken link. The accepted answers can provide values to questioners to solve their problems. However, because viewers cannot fully understand the answer with broken links, the accepted answers with broken links cannot provide values to the future viewers.

troduced in Aug 2008 (the month when the Stack Overflow website was established) are broken. We structure our study by answering the following research questions:

1) **What are the intended roles of the broken links on Stack Overflow?**
65% of the broken links in our sampled questions are used to show examples, e.g., code examples. 70% of the broken links in our sampled answers are used to provide supporting information, e.g., explaining a certain concept and describing a step to solve a problem.

2) **Is there any significant difference of popularity between broken posts and normal posts?**
Only 1.57% of the broken posts are highlighted as such by viewers in the posts' comments. Only 5.8% of the broken posts repaired the broken links. Viewers cannot fully rely on the vote scores to detect broken links, as broken links are common across posts with different vote scores.

3) **Which websites are referenced by broken links the most on Stack Overflow?**
50% of the broken links reference to the top 0.3% websites ordered by the number of the broken links referencing to them. The websites that host the resources maintained by their users are referenced by broken links the most, e.g., github.com.

4) **Are the posts and comments associated with particular tags more likely to have broken links than others?**
55.4% of the broken links are referenced in the posts and comments that are marked with the top 10 tags ordered by the number of the broken links associated with them. The posts and comments related to the web technologies, i.e., JavaScript, HTML, CSS, and jQuery, are associated with more broken links.

Based on our findings, we provide actionable suggestions for Stack Overflow moderators, users, and future researchers. For example, we encourage Stack Overflow to detect and mark the questions with broken links to notify viewers of the broken links. For Stack Overflow users, we recommend Stack Overflow users to post the code in the code blocks or Stack Snippets as much as possible, rather than the external code websites, e.g., github.com. For future researchers, we suggest they could repair broken links based on the revisions of links. We publish the scripts, data, coding guides, etc. on Zenodo (i.e., a preserved archive).[2]

Paper Organization: The remainder of the paper is organized as follows. Section 2 provides the background information of Stack Overflow and describes the related work about the knowledge sharing in software engineering and the broken links across the Internet. Section 3 details our approach to collect and process the data which are used in our study. Section 4 presents preliminary studies on the broken links on Stack Overflow. Section 5 presents our research findings by answering the aforementioned four research questions. Section 6 provides actionable suggestions based on our findings and acknowledges some of the key threats to the validity of our study. Finally, Section 7 concludes our study and proposes potential future work.

2 Background and Related Work

In this section, we present the background information of Stack Overflow and discuss the related work about the knowledge sharing in software engineering and broken links across the Internet.

2.1 Studies on Stack Overflow

Stack Overflow is a well-known online Q&A site to answer programming-related problems. Questioners can post questions that include textual descriptions [6]. Each question may receive multiple answers [7]. Answers contribute the solutions to the crowdsourced knowledge on Stack Overflow. Besides, registered viewers can comment under each post to notify the owner of the post for a clarification. Many researchers focused on characterizing the Stack Overflow questions, answers and comments [8], [9]. Saha et al. investigated why questions remain unanswered and concluded that the majority of them were due to low interest in the community [8]. Linares-V'asquez et al. investigated the relationship between API changes in Android SDK and developers' reactions to those changes on Stack Overflow [9]. They observed that Android developers usually have more questions when the behavior of APIs is modified. Zhang et al. investigated the obsolete knowledge on Stack Overflow and observed that more than half of the obsolete answers were probably already obsolete when they were posted [5]. Zhang et al. observed that comments on Stack Overflow can be leveraged to improve the quality of their associated answers [10].

To capture the topics with which a question is associated, questioners need to specify the tags into well-defined categories when they post a question [11]. Each question can have at most five tags and must have at least one tag. The tags can facilitate dispatching questions to the potential

2. https://zenodo.org/record/4683732

users who are interested in it. Many researchers focused on the topics of the Stack Overflow questions [12]–[16]. They investigated the topic trends across the whole Stack Overflow [12], or in specific communities, e.g., mobile [13] and web [14]. Xia et al. proposed an approach to recommend tags to software information sites [17]. Xu et al. designed a tool to recognize semantically relevant knowledge units on Stack Overflow [16].

Users can include code snippets and other references (e.g., links or images) to enrich their posted questions [18]. Many researchers investigated how to utilizing the knowledge hosted on Stack Overflow to help with software engineering as well [19]–[21]. Chen et al. used the code blocks from Stack Overflow to detect defective code fragments in developers' source code [19]. Cai et al. and Huang et al. used the knowledge hosted on Stack Overflow to recommend APIs [20], [21].

To ensure the quality of the crowdsourced knowledge on Stack Overflow, Stack Overflow propose a gamification system. Questioners can accept the answers that can solve their questions [22], i.e., can provide instant values to the developers who proposed questions on Stack Overflow. Registered viewers could benefit from the accepted answers to learn the best way to solve the problems. Registered viewers also can vote up the questions and answers that are useful to them, i.e., can also provide long-lasting values to the developers who encounter similar problems that are already asked on Stack Overflow [23], [24]. By posting high-quality questions and answers, and suggesting reasonable edits, users can earn points to increase their reputations. Considering the success of Stack Overflow, many researchers investigated the benefits of the gamification mechanisms [25]–[30]. Anderson et al. designed a tool to determine which questions and answers are likely to have long-lasting value, and which ones are in need of additional help from the community [27]. Pal et al. investigated the evolution of experts on the Stack Overflow community and pointed out how expert users differ from ordinary users in terms of their contributions [28]. Hanrahan et al. developed indicators for difficult problems and experts [29]. They examined how complex problems are handled and dispatched across multiple experts.

Any user can suggest edits to revise a question's title, body, and tags, or an answer's body [18]. Suggested edits from the original questioners and answerers will be applied immediately, as well as from users who have more than 2,000 reputation points (2k users). Other users' suggested edits will be reviewed by the 2k users to decide whether to be applied or not. Many researchers investigated the collaborative editing on Stack Overflow [30]–[33]. Li et al. observed that the benefits of collaborative editing on questions and answers outweigh its risks [30]. Wang et al. observed that 25% of the users did not make any more revisions once they received their first revision-related badge [31]. Chen et al. developed an edit-assistance tool to identify minor textual issues in posts and recommending sentence edits for correction [32]. They also developed a Convolutional Neural Network-based approach to learn editing patterns from historical post edits for identifying the need for editing a post [33]. To characterize what do users (e.g., questioners, answers, and commenters) do after the observations of broken links, in this paper, we analyze the applied edits in this paper.

2.2 Link Sharing in Software Engineering

Researchers investigated the links in Stack Overflow. Ye et al. investigated the **internal links** (i.e., links that reference to the resources hosted within the Stack Overflow website) to analyze the evolution of the knowledge network that is connected by the internal links [34]. Gómez et al. [35] investigated the **external links** (i.e., links that reference to the resources hosted outside the Stack Overflow website) from the link types, website types, and the most referenced links and websites perspectives. Baltes et al. analyzed the purpose of the links that reference to documentation websites on Stack Overflow, e.g., pointing to API documentation and concept descriptions on Wikipedia for background readings [36]. Correa et al. investigated the role and impact of Stack Overflow in issue tracking systems [37]. They observed that the average number of comments posted in response to bug reports is less when Stack Overflow links are presented in the bug report. Wang et al. revealed the links between the Android issues in bug tracking systems and Stack Overflow posts by integrating the semantic similarity between Android issues and Stack Overflow posts [38].

Researchers also investigated the links between software engineering artifacts. For example, Rath et al. investigated the inter-linking of commits and issues in open source projects and observed that among six large projects, 60% of the commits are linked to issues [39].

On utilizing web resources, Xia et al. listed the frequency and difficulty of the different web search tasks performed by developers [2]. Rahman et al. proposed a novel IDE-based web search that exploits three reliable web search engines (e.g., Google, Bing, and Yahoo) and a programming Q&A site (i.e., Stack Overflow) through their API endpoints [1]. Gao et al. developed an automatic web resources linking technique to linkify entity mentions to relevant official documentation in Stack Overflow [40].

Similar to the aforementioned studies, our work investigates knowledge dissemination in software engineering. We focus on the broken links on Stack Overflow, rather than all the links, to study the broken link-sharing activities.

2.3 Broken Links Across the Internet

The HTTP response status code (i.e., response code) represents the result of the response of the website serve to the request of the link. The response code is a three-digit integer. The first digit of the status-code defines the class of response. For example, 2xx response code represents the action requested by the client is received, understood, and accepted; 4xx response code represents that the request contains bad syntax or cannot be fulfilled; 5xx response code represents that the server failed to fulfill an apparently valid request [41].

Habibzadeh et al. examined the prevalence of the broken links in academic literature [42]. They found that ranging 35.4% to 68.4% of the links in different journals are broken links. Fetterly et al. observed that about one link out of every 200 broke each week on the Web [43]. Koehler et al. observed that the links could have dramatically different half-lives [44], the links selected for publication appear to have greater longevity than the average links. A 2015 study by Weblock analyzed more than 180,000 links from references in the full-text corpora of three major open-access publishers. This study found that 24.5% of the studied links are broken[3]. McCown et al. observed that half of the links cited in D-Lib Magazine articles were active 10 years after publication [45]. Hennessey et al. analyzed nearly 15,000 links in abstracts from Thomson Reuters's Web of Science citation index [46]. They observed that the median lifespan of web pages was 9.3 years, and just 62% were archived. Klein et al. observed that one out of five Science, Technology, and Medicine articles suffering from reference rot, meaning it is impossible to revisit the web context that surrounds them after their publication [47]. Zeng et al. observed that most resources linked in biomedical articles disappear in 8 years [48].

Different from the aforementioned studies, our study inspects the broken links in a popular software engineering related Q&A websites, i.e., Stack Overflow. Stack Overflow host a large collection of knowledge for developers to solve their programming-related problems. Inspecting broken links on Stack Overflow could enable us to understand the broken links problems in the software engineering field.

3 Experiment Setup

In this section, we present the data collection steps that we used to extract the links from the SOTorrent dataset and test the availability of links.

3.1 Data Collection

Links on Stack Overflow can reference to the resources that are scattered across the Internet. The links with *stackoverflow.com* root domain reference to the resources hosted within the Stack Overflow websites. In contrast, the links without *stackoverflow.com* root domain reference to the resources hosted outside the Stack Overflow websites. In this paper, we investigate the availability of the links that reference to the resources hosted outside the Stack Overflow websites. **We do not consider the availability of the links that reference to the resources hosted within the Stack Overflow websites** because these links are maintained by Stack Overflow. For example, the Stack Overflow moderators are aware of the deleted questions and have taken some actions, e.g., displaying the questions that are similar to the deleted question [49].

We use the SOTorrent dataset[4] [50], [51] to obtain the links in the **text** of posts (i.e., questions and answers) and comments on Stack Overflow. SOTorrent dataset is based on the official Stack Overflow data dump that hosts the website data from Jul. 31, 2008 to Jun. 2, 2019. Baltes et al. extracted and identified the text blocks and code blocks from all versions of posts from the Stack Overflow data dump table `PostHistory` and stored these blocks into table `PostBlockVersion` [50], [51]. Baltes et al. collected the links in **the text blocks** in all versions of Stack Overflow posts using a regular expression and then stored these links into table `PostVersionUrl`. They also extracted the links in Stack Overflow comments with a regular expression and then store these links into table `CommentUrl`. We encourage readers to read their work for the full details of the data collection process of the SOTorrent dataset.

We extract the links **in the text of posts and comments** from the SOTorrent table `PostVersionUrl` and table `CommentUrl`. Links in the text of posts and comments are used to reference to resources. We identify who shared the link and when the link was shared from table `PostHistory` for Stack Overflow posts and table `Comments` for comments. We identify the links that reference to the resources hosted outside the Stack Overflow websites using their root domain as we mentioned above. As a result, we finally obtain 12,446,901 links in Stack Overflow history, and 11,926,354 of them are in the latest version of Stack Overflow posts and comments (i.e., 520,574 links are not shared currently).

3.2 Link Availability Test

We perform the link availability test using Scrapy[5]. Scrapy is an open-source web crawling and web scraping framework. To identify the broken links, we obtain the HTTP response status code (i.e., response code) that is returned to the request for the resource referenced by the link. To avoid the IP being banned from the website, we obtain a list of proxies from Free Proxy List[6] and make different requests using different proxies. For the same website, we set a 15 seconds delay for different requests. To reduce the bandwidth requirement, we only request the header of the response. To mitigate the intermittent behavior, we perform one link availability test using an elastic compute service located in Virginia, U.S. in Dec 2019. For the links that are not responded with 2xx response code, in Jan 2020, we perform another link availability test using an elastic compute service located in Singapore. We identify the broken links that are not responded with 2xx response code in both trials.

4 Preliminary Studies on the Broken Links on Stack Overflow

In this section, we present a series of preliminary studies related to the broken links on Stack Overflow, including the prevalence of broken links, the response code of the broken links, and the broken links that were posted per month.

4.1 Prevalence of Broken Links on Stack Overflow

14.2% (i.e., 1,687,995) of the links on Stack Overflow are broken links. 13.5% (i.e., 2,156,095) of the posts and comments with links have broken links. Also, 10.8% (i.e., 2,493,328) of the occurrences of the links are broken links. Note that one link can be shared multiple times in a post or comment. Such a large proportion of broken links would downgrade the overall quality of the Stack Overflow. However, it is still unclear what are the characteristics of the broken links on Stack Overflow. This motivates us the

3. https://web.archive.org/web/20160304081204/https://weblock.io/report?id=all
4. https://zenodo.org/record/3255045#.XYWaMyh3iUk
5. https://scrapy.org/
6. https://free-proxy-list.net/#list

further investigate the broken links on Stack Overflow. By doing so, we could shed lights for future directions and provide recommendations for practitioners and researchers to address the broken links issue.

4.2 Response Code of the Broken Links on Stack Overflow

To have a basic understanding of the broken links on Stack Overflow, we investigate the response code of the broken links. To do so, we group the broken links according to the response code and count their corresponding numbers.

Table 1 presents the top 10 response code of the broken links on Stack Overflow [52]. **50.7% (i.e., 856,017) of the broken links are responded with the 404 response code.** The 404 response code corresponds to a resource *Not Found*; however, it does not indicate whether unavailability is temporary or permanent. In contrast, the 410 response code explicits that the resource is likely to be permanently removed [52]. On Stack Overflow, only 0.6% of the broken links are responded with the 410 response code. We encourage the Stack Overflow moderators to remove the broken links that are responded with the 410 response code.

The 403 response code is another common response code among the broken links on Stack Overflow. This response code indicates that the access to the resource requires authentication. However, external visitors do not have the access. Following the 404 and the 403 response code, DNS Lookup Error is the third and TCP Timed Out Error is the fourth most common status code for broken links. One possible reason is that the website servers of these broken links fail to provide any response.

4.3 Trendlines of Broken Links on Stack Overflow

Here, we would like to investigate whether the links that were posted earlier on Stack Overflow are more likely to be broken. By doing so, we can better understand whether the broken links on Stack Overflow is time-related.

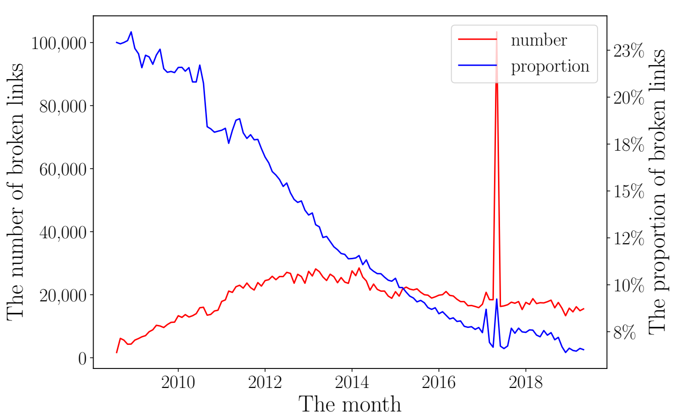


Fig. 2: The numbers of broken links and the proportions of broken links among the links that were posted per month. This figure shows that the links that were posted earlier are more likely to be broken.

Figure 2 shows the numbers of broken links and the proportions of broken links among the links that were posted per month. The mean and median number of broken links included in Stack Overflow posts in a month is 19,033 and 18,751 respectively. The proportions of broken links among the links that were posted per month and the month the links are posted are significantly correlated with Pearson's correlation coefficient = -0.97 (p-value < 0.05). 22.9% of the links that were posted in Aug 2008 (the month when the Stack Overflow website was established) are broken. **Links that were posted earlier are more likely to be broken.** This indicates that broken links on Stack Overflow are time-related.

Stack Overflow posts made in May 2017 have a total of 103,411 broken links (the spike in Figure 2). The number of broken links posted in May 2017 is 5.43 times greater than the average number of broken links that were posted per month. We manually check the links that were posted in May 2017 and observe that 78.8% (i.e., 882,362) of the links were posted by the URL Rewriter Bot. As is indicated in `meta.stackexchange.com` (i.e., the website for the meta-discussion of the Stack Exchange family of Q&A websites), the URL Rewriter Bot is used by Stack Overflow to update the schema of the links, i.e., replacing HTTP with HTTPS for security and privacy concern without checking their validity[7,8,9]. **However, 87,086 broken links (i.e., 84.2% of the broken links posted in May 2017) were posted by the URL Rewriter Bot.** One possible reason is that widely applying the effective approach to resolve the link security problem is much simpler than widely maintaining the broken links on Stack Overflow. For example, to resolve the link security problem, Stack Overflow proposed a tool to automatically replace HTTP with HTTPS.[10] To resolve the broken links to images, Stack Overflow used a crowdsourced approach to update broken image links (i..e., require human intelligence to manually update broken links).[11]

5 Findings

In this section, we present the results of our empirical study that answer the four research questions related to the broken links on Stack Overflow. More specifically, we analyze the intended roles of the broken links in Stack Overflow questions, answers, and comments. We investigate the impact of the broken links on the crowdsourced knowledge on Stack Overflow. We also characterize the tags that are associated with broken links and the websites that are referenced by broken links.

5.1 What are the Intended Roles of the Broken Links on Stack Overflow?

Motivation: In Section 4.1, we have identified that broken links are prevalent on Stack Overflow. Previous work observed different roles of the links on Stack Overflow [34]. However, it is still unclear what the *intended* roles of the broken links in knowledge dissemination are. More

7. https://meta.stackexchange.com/q/291947/
8. https://meta.stackoverflow.com/q/345012/
9. https://nickcraver.com/blog/2013/04/23/stackoverflow-com-the-road-to-ssl/
10. https://nickcraver.com/blog/2017/05/22/https-on-stack-overflow/
11. https://meta.stackexchange.com/q/291976

TABLE 1: Top 10 response codes returned for the broken links. This table shows that the 404 error is the main error code for the broken links on Stack Overflow. # = number of broken links returning the corresponding status code. % = percentage of broken links returning the corresponding status code.

	HTTP Status Code	Explanation	#	%
1	404	The requested resource could not be found currently.	856,017	50.7%
2	403	User not having the necessary permissions for the resource.	220,261	13.0%
3	DNSLookupError	DNS lookup failed.	213,684	12.7%
4	TCPTimedOutError	TCP connection timed out.	66,940	4.0%
5	405	The request method is not supported for the requested resource.	46,648	2.8%
6	503	The server cannot handle the request.	43,515	2.6%
7	500	The server failed to fulfill the request.	30,873	1.8%
8	400	The server cannot or will not process the request due to an apparent client error.	19,030	1.1%
9	401	Authentication is required and has failed or has not yet been provided.	12,230	0.7%
10	410	The resource requested is no longer available and will not be available again.	10,148	0.6%

specifically, we would like to understand the intended roles of the broken links in Stack Overflow questions, answers, and comments respectively.

Approach: To analyze the intended roles of the broken links in knowledge sharing, we follow Ye et al.'s work, where they analyzed the roles of sharing internal links on Stack Overflow [34]. Ye et al. observed there are five general roles of link sharing in Stack Overflow, i.e., 1) reference information for problem-solving, 2) reference existing answers, 3) reference visited but not helpful web pages, 4) recommend related information, and 5) others. We randomly sampled a statistically sample of 384 posts and comments with broken links from questions, answers, and comments, respectively (i.e., 1,152 posts and comments broken links in total), using a 95% confidence level with a 5% confidence interval. We want to understand the characterize the intended roles of broken links in Stack Overflow questions, answers, and comments, respectively. To label the intended roles of the broken links, we manually performed a lightweight open coding process to check the discussion context where the broken links are referenced. This process involves 3 phases and is performed by the first two authors of this paper:

- Phase I: We randomly selected 100 broken links from the sampled 1,152 broken links. The first two authors used the coding schema used by Ye et al.'s work to categorize the selected 100 broken links collaboratively [34]. During this phase, the coding schema of the intended roles of the broken links on Stack Overflow was revised and refined. We performed the refinement because we observed that the intended role of many broken links is referencing information for problem-solving, we divided the intended role of "reference information for problem-solving" into two intended roles, i.e., providing working examples and providing supporting information.
- Phase II: The first two authors applied the resulting coding schema of Phase I to categorize the remaining 1,052 broken links independently. They were instructed to take notes regarding the deficiency and ambiguity of the coding schema for categorizing certain broken links. The inter-rater agreement (Cohen's kappa) of this stage is 0.69, indicating that the agreement level is substantial [53].
- Phase III: The first two authors discussed the coding results obtained in Phase 2 to resolve the disagreements. We did not invite others because all the disagreements were resolved during the discussion. For example, for the broken link in an answer[12] to whether there is a ready Ajax extender or a JQuery functionality to implement search textbox,

For the full code check out the following post: http://www.simplygoodcode.com/2013/08/placing-text-and-controls-inside-text.html[13]

The first author considered the intended role of this broken link is to provide supporting information because the post owner encourages viewers to check out the *post* for more details. The second author considered the intended role of this broken link is to show code examples as "the full code can be found in the broken link". We finally consider the intended role of this broken link is to provide supporting information as the broken link is a personal blog that records the approaches to solve programming related problems. The first two authors maintained the coding schema to resolve schema deficiencies and ambiguities. For example, Ye et al. observed that referencing the existing answers can be a "confirmed duplicate" if there is a [duplicate] marker at the end of the question title. However, Stack Overflow does not confirm the duplicate to the web page on another website. Therefore, we do not consider "confirmed duplicate" as a kind of "existing answers" in our coding schema. At the end of Stage 3, we obtained the final coding schema and the final coding results of the sampled 1,152 broken links.

Results: Table 3 shows the prevalence of the broken links in all types of Stack Overflow posts and comments. We find that the proportions of questions and comments that have

12. https://stackoverflow.com/q/18369005/
13. http://www.simplygoodcode.com/2013/08/placing-text-and-controls-inside-text.html
14. http://wp.matthewwood.me/
15. https://stackoverflow.com/q/31938291/
16. http://mattiasholmqvist.se/2010/03/building-with-tycho-part-2-rcp-applications/
17. https://stackoverflow.com/q/18550820/
18. https://www.quora.com/Java-When-we-concatenate-two-strings-using-the-+-operator-will-the-resulting-string-be-stored-in-the-string-literal-pool-or-not?share=1
19. https://stackoverflow.com/q/44050772/
20. http://public.kitware.com/Bug/bug_revision_view_page.php?-rev_id=958
21. https://stackoverflow.com/q/15159722/
22. https://msdn.microsoft.com/en-us/windows/uwp/globalizin-g/put-ui-strings-into-resources
23. https://stackoverflow.com/q/40120304/
24. http://msmvps.com/blogs/jon_skeet/archive/2009/02/17/an-swering-technical-questions-helpfully.aspx
25. https://stackoverflow.com/q/12838984/

TABLE 2: Intended roles for broken links. This table shows that most of the broken links in questions are used to show examples, and most of the broken links in answers are used to provide supporting information on users' claims.

Intended Role	Definition	Example	% Questions	% Answers	% Comments
Working Example	Provide working examples, e.g., code snippets.	if you go to wp.matthewwood.me[14] and click through the links you will see what I mean ... [15]	65%	22%	49%
Supporting Information	Explain a certain concept, approach of (sub)step to solve the questions, background knowledge, or the link sharer's claim.	However I followed the tutorials where I create a plugin, feature ... http://mattiasholmqvist.se/2010/03/building-with-tycho-part-2-rcp-applications/[16] seems a bit out of date.[17]	22%	70%	44%
Existing Answers	Reference existing answers in a Q&A websites.	Interesting read on the same topic - quora.com/...[18]...[19]	0%	0%	1%
Visited Webpages	Reference visited "search and research" web pages that cannot solve the problem.	Things I've already read ... CMake bug report 0013765[20] - this[21]	13%	0%	2%
Related Information	Recommend related information that does not directly answer the question.	I think so, more details please reference this article[22][23]	0%	8%	3%
Others	Suggest how to good asks.	And what else? Could you add some more information. Check this metaSO question and Jon Skeet: Coding Blog[24] on how to give a correct answer.[25]	0%	0%	1%

TABLE 3: Prevalence of the broken links in Stack Overflow questions, answers, and comments. This table shows that questions and comments have higher proportions of broken links, and answers contribute the largest number of broken links.

	# Posts	% Posts	# Links	% Links	# Occurrences	% Occurrences
Questions	620,837	16.8%	635,062	14.4%	752,414	13.9%
Answers	905,964	11.0%	670,145	11.0%	1,061,838	8.5%
Comments	641,448	13.7%	556,417	18.5%	679,076	13.2%

broken links among all questions and comments are 1.52 times and 1.24 times higher than for answers respectively. The proportions of broken links among all links in questions and comments are 1.31 times and 1.68 times higher than for answers respectively. The proportions of the occurrences of broken links among the occurrences of all links in questions and comments are 1.63 times and 1.55 times higher than answers respectively. The above indicates that **questions and comments have higher proportions of broken links than answers.**

Moreover, the number of answers that have broken links is 1.46 times and 1.41 times higher than for questions and comments respectively. The number of broken links in answers is 1.06 times and 1.20 times higher than for questions and comments respectively. The number of the occurrences of broken links in answers is 1.41 times and 1.56 times higher than for questions and comments respectively. This shows that **answers contribute more to the absolute numbers of broken links than questions and comments.** 10.1% (i.e., 297,305) of the links in accepted answers are broken links. 10.7% (i.e., 338,705) of the accepted answers have broken links. 44.4% (i.e., 297,305) of the broken links in answers are posted in accepted answers. This shows that **broken links are common in the accepted answers. The answers with broken links could have solved questioners' problems when the links were posted, i.e., before the links were broken.**

Table 2 shows the intended roles of the broken links in Stack Overflow questions, answers, and comments, respectively. **We observe that 65% of the broken links in our sampled questions are used to show examples, e.g., the demos of the tasks and the code examples written by the post owners.** Questioners explain the tasks with these links when they post their questions. These links may reference to test cases, demos, or the development versions of a software. After the problem was resolved, the questioners removed the example from the link. This practice cause the link to be broken. However, without these links, the following viewers cannot tell whether the question is similar or even identical to the problems they are facing with. Because the following viewers cannot benefit from the questions, the broken links in questions can lead to the questions with broken links to be useless. For example, in a comment to a question[26]:

> *Do you still have that code? If so, please edit it in. Your Dropbox link is dead so the question is useless now.*

The broken link[27] is the test page to show an example of the questioner's problem. Viewers cannot understand the question without the example hosted in the broken link. We suggest that users should not remove the examples in links.

In our sampled answers, 70% of the broken links are used to provide supporting information, e.g., a certain concept and a step to solve a problem. Such information is provided by the Stack Overflow communities to solve programming-related questions. For example, in a comment to an accepted answer[28]:

26. https://stackoverflow.com/q/7589262/
27. http://dl.dropbox.com/u/3085200/canvasTest/index.html
28. https://stackoverflow.com/q/15488527/15489656

The link is dead. Could you fix it? I'm very much interested in a solution to this problem.

The viewer complains that he cannot fully understand the answer with the broken link. The broken link[29] explains the substep to solve the problem. This leads to the answers useless and damage the reputation of the Stack Overflow.

94% of the broken links in our sampled comments can be used in problem-solving. Table 2 shows that the broken links in comments can be used in problem solving to provide working examples (i.e., 49%), supporting information (i.e., 42%), existing answers (i.e., 1%), and visited but not helpful knowledge (i.e., 2%), etc. This finding is consistent with Zhang et al.'s work [10], where they observed that comments on Stack Overflow can be leveraged to improve the quality of the associated answers [10]. However, we observe that broken links are common in comments. We suggest the Stack Overflow moderators should pay attention to the maintenance of comments as well.

> Questions and comments have higher proportions of broken links than answers. Stack Overflow answers contribute broken links the most compared with questions and comments. 65% of the broken links in our sampled questions are used to show examples, e.g., code examples. 70% of the broken links in our sampled answers are used to provide supporting information, e.g., explaining a certain concept and describing a step to solve the problem.

5.2 Is there any significant difference of popularity between broken posts and normal posts?

Motivation: In Section 2.1, we introduce the mechanisms of Stack Overflow. For example, the registered viewers can vote up the crowdsourced knowledge (i.e., questions and answers) that are useful to them, i.e., can provide values to the developers who encounter similar problems that are already asked on Stack Overflow [23], [24]. Users can suggest edits to maintain the Stack Overflow posts [18]. However, it is still unclear whether there is a significant difference of popularity between broken posts and normal posts? By doing so, we could better understand the characteristics of the broken links on Stack Overflow at a large scale case.

Approach: Registered viewers can comment under each post to notify the owner of the post for a clarification [54]. To capture an overview of how often **viewers notify the posts owners of the broken links**, we extract **the comments indicating the notification of broken links in posts**. More specifically, we identify 2,727,969 comments to the broken posts. To identify the comments indicating the notification of broken links in posts, we use the keywords that indicate broken links in previous work [42]–[46], wordnet[30], and other online resources[31]. For example, "broken", "404", "unavailable", "rot", "inaccessible", "dead", and so on, can represent the keyword "broken"; "url", "reference", "citation", and so on, can represent the keyword "link". Because there are various reasons why links are updated, e.g., updating the obsolete knowledge, we do not use the keyword "update" to identify the comments indicating the notification of broken links. **Finally, we obtain a total of 27,261 comments indicating the notification of broken links in 25,443 posts.** 2,700,708 comments are not identified as the comments indicating the notification of broken links. To check the precision of identifying the comments indicating the notification of broken links, we randomly sampled a statistically representative sample of 379 comments from 27,261 identified comments using a 95% confidence level with a 5% confidence interval. We find that our heuristics achieves a precision score of 0.88 as 42 comments that are false positive (i.e., not the comments indicating the notification of broken links). Similarly, to check the recall of identifying the comments indicating the notification of broken links, we randomly sampled a statistically representative sample of 379 comments from 2,700,708 comments using a 95% confidence level with a 5% confidence interval. We find that there is no comment indicating the notification of broken links.

To capture an overview of how often users removed the broken links, we compare the number of posts that do not have broken links currently with the number of posts that had broken links in history.

To investigate the difference of popularity between the broken posts and the normal posts, we extract the vote scores of posts and the view count of questions from the Stack Overflow data dump table `Posts`. We extract the date of each vote from the Stack Overflow data dump table `Votes`. We investigate the differences of the **accumulative vote scores** between the broken posts and the normal posts. To do so, for Stack Overflow questions with links, we compare the vote scores of the questions with broken links with the questions without broken links in the same range of view counts. For Stack Overflow answers with links, we exclude the questions that only have one answer. Then we compare the ranks of the answers with broken links among all answers to a specific question with those of the answers without broken links in terms of the vote scores.

Moreover, Stack Overflow provides additional popularity metrics for questions, i.e., favorite count and view count. Stack Overflow allows users to bookmark questions by clicking the *Favorite* icon in questions.[32] By doing so, users could easily visit again to check updates in the future. The higher favorite counts to a question indicate a larger number of users that would like to track the question. View count records the number of visits to Stack Overflow questions. We extract the view count and the favorite count of questions from the Stack Overflow data dump table `Posts`. Then, we check the proportion of questions with broken links in different view count ranges and favorite count ranges.

As registered viewers can vote up the posts that are useful to them, viewers would expect to receive more help from the posts with higher vote scores. To do so, we check the proportion of broken links among the links in the questions and the answers with different vote score ranges.

To analyze whether the use of broken links is associated with the user reputation, we extract the id and the name of

29. http://home.roadrunner.com/ hinnant/stack_alloc.html
30. http://wordnetweb.princeton.edu/perl/webwn
31. https://en.wikiredia.com/wiki/Wikipedia:Link_rot
32. https://meta.stackexchange.com/q/23670/

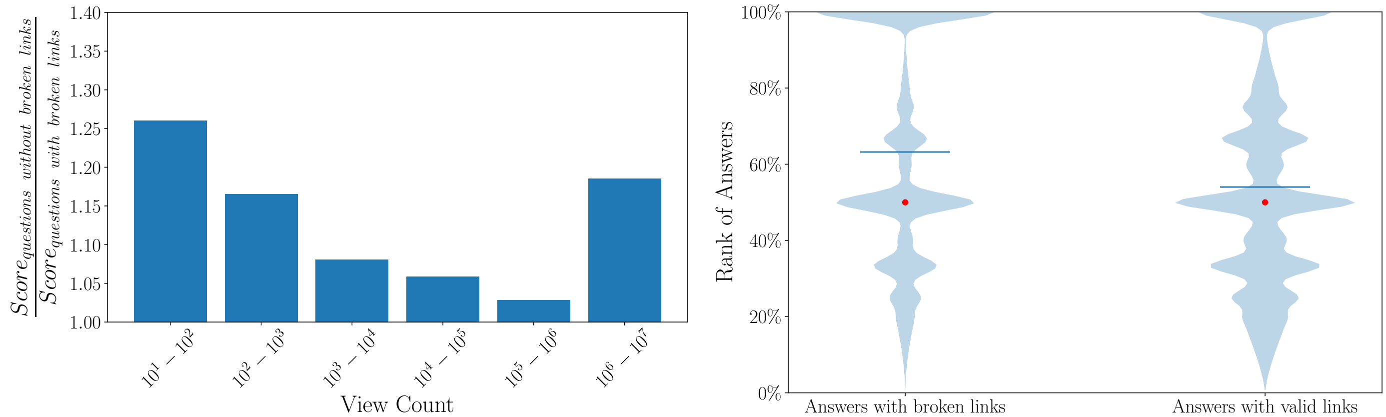
the users who posted the links from the Stack Overflow data dump table `PostHistory`. We extract the user reputations from the Stack Overflow data dump table `Users`. Then we check the proportion of broken links among the links posted by users with different reputation ranges.

Results: Only 5.8% (i.e. 103,792) of the broken links were removed. 5.6% (i.e., 90,606) of the broken posts removed the broken links. 6.7% (i.e., 164,404) of revisions that changed the links in posts removed broken links. This shows that broken links attract limited attention on Stack Overflow. **1.57% (i.e., 25,443) of the broken posts are notified of the broken links via one or more comments (i.e., notified posts). 14.3% (i.e., 3,648) of the notified posts removed the broken links.** The proportion of the notified posts that removed broken links among all notified posts is 2.47 times larger than the proportion of the posts that removed broken links among the broken posts. This shows that when notified of the broken links, users are more likely to repair the broken links. We suggest Stack Overflow could design a mechanism to notify broken links in posts.

For Stack Overflow questions, Figure 3a shows the relations between the accumulative view counts and the accumulative vote scores. To check whether the differences in the number of vote scores per view count are statistically significant between the questions with broken links and the questions without broken links, for the questions with the different ranges of view counts, we perform a Mann Whitney test [55]. The null hypothesis is that there is no difference between the questions with broken links and the questions without broken links in terms of the number of vote scores per view count in different ranges of view counts. As a result, the difference between the questions with broken links and the questions without broken links is significant (p-values < 0.05). We then calculate Cliff's delta to measure the effect size [56]. Cliff's delta is a non-parametric effect size measure that can evaluate the amount of difference between two groups. Romano et al. define an absolute delta of less than 0.147, between 0.147 and 0.33, between 0.33 and 0.474 and above 0.474 as "Negligible", "Small", "Medium", "Large" effect size, respectively [57]. As a result, the difference between the questions with broken links and the questions without broken links in terms of the number of vote scores per view count in different ranges of view counts is medium (Cliff's delta is between 0.33 and 0.474). More specifically, we observe that **for the questions with the same range of view counts, questions without broken links are associated with higher vote scores compared with the questions with broken links**. When viewers browse the Stack Overflow website, they would encounter the questions with broken links and the questions without broken links. Finally, they vote on the questions without broken links to express the usefulness of the questions.

Figure 3b shows the ranks of the answers with broken links and the ranks of the answers without broken links among all answers to a specific question in terms of the vote scores. To check whether the differences in the vote score ranks of the answers to a specific question are statistically significant between the answers with broken links and the answers without broken links, we perform a Mann Whitney test [55]. The null hypothesis is that there is no difference between the answers with broken links and the answers without broken links in terms of the vote score ranks of the answers to a specific question. As a result, the difference between the answers with broken links and the answers without broken links is significant (p-values < 0.05). We then calculate Cliff's delta to measure the effect size [56]. As a result, the difference between the answers with broken links and the answers without broken links in terms of the vote score ranks of the answers to a specific question is small (Cliff's delta is between 0.147 and 0.33). **This shows that for the answers to the same questions, answers without broken links are associated with higher vote scores compared with the answers with broken links.** When viewers browse the answers to a specific question, they would encounter the answers with broken links and the answers without broken links. Finally, they vote on the answers without broken links to express the usefulness of the answers.

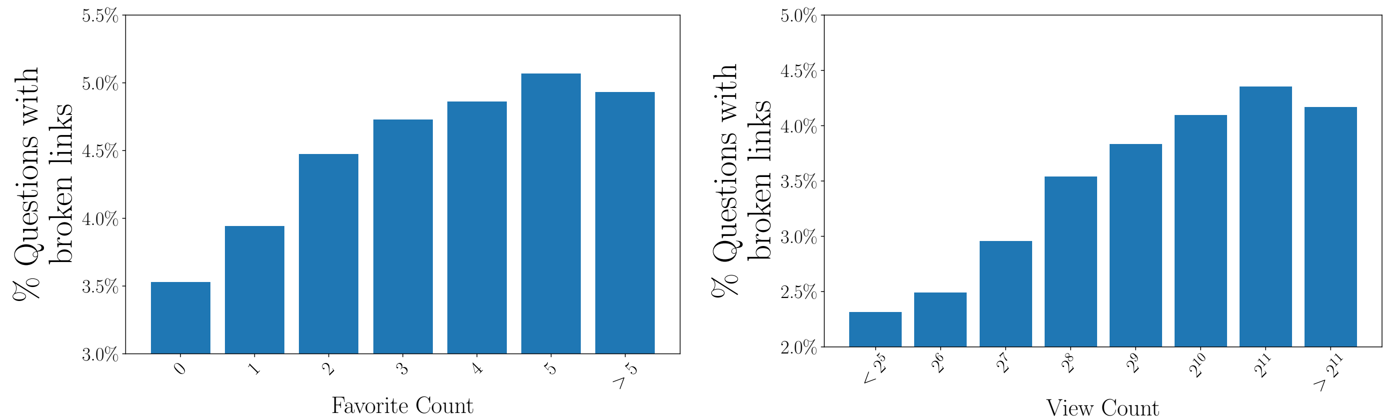
Broken links are more common in questions with higher view counts and higher favorite counts. Figure 4 shows the distribution of the proportions of questions with broken links in different view count ranges and favorite count ranges. For Stack Overflow questions with different ranges of favorite counts, the proportions of questions with broken links increase from 3.6% (for the questions with none of favorite counts) to 5.1% (for the questions with 5 favorite counts). This indicates that for the questions across different favorite counts, the proportions of the posts without broken links are higher than the proportion of posts with broken links. More specifically, we observe that the proportions of questions with broken links and the favorite count of questions are significantly correlated with Spearman's rank correlation coefficient = 0.96 (p-value < 0.05) [58]. This indicates that users could encounter more broken links in the questions with more favorite counts. For questions with different ranges of view count, the proportions of questions with broken links increase from 2.4% (for the questions with fewer than 2^5 view counts) to 4.3% (for the questions with view count more than 2^{10} and less than 2^{11}). This indicates that for the questions across different view counts, the proportions of the posts without broken links are higher than the proportion of posts with broken links. More specifically, we observe that the proportions of questions with broken links and the view count of questions are significantly correlated with Spearman's rank correlation coefficient = 0.98 (p-value < 0.05) [58]. This indicates that users would be more likely to encounter broken links in the questions that are viewed more. The underlying reason could be that questions with higher view counts and higher favorite counts can be posted earlier. In Section 4.3, we observe that links that were posted earlier are more likely to be broken. To check whether questions with higher favorite counts are posted earlier, we calculate Spearman's rank correlation coefficient between the favorite count of questions and the creation time of questions. As a result, we observe that the favorite count of questions and the creation time of questions are significantly correlated with Spearman's rank correlation coefficient = 0.27 (p-value < 0.05) [58]. This indicates that the favorite count and the creation time of questions have a weak correlation [59]. One possible reason is that users might use other approaches (e.g., browsers) to bookmark questions due to



(a) The ratios of the score of questions without broken links to the score of questions with broken links in different view count interval.

(b) The ranks of the answers with broken links and the answers without broken links among all answers to a specific question in terms of the vote scores.

Fig. 3: The vote scores on the questions and answers with or without broken links.



(a) Favorite Count

(b) View Count

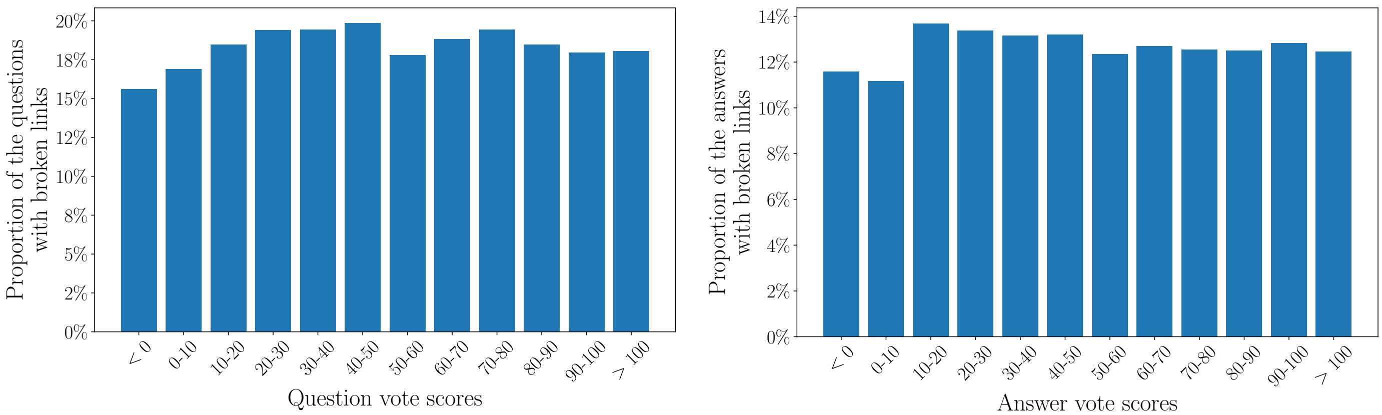
Fig. 4: The proportions of questions with broken links in different view count ranges and favorite count ranges. These figures show that broken links are more common in questions with higher view counts and higher favorite counts.

privacy concerns.[33] To check whether questions with higher view counts are posted earlier, we calculate Spearman's rank correlation coefficient between the view count of questions and the creation time of questions. As a result, we observe that the view count of questions and the creation time of questions are significantly correlated with Spearman's rank correlation coefficient = 0.53 (p-value $<$ 0.05) [58]. This shows that questions with higher view counts are associated with earlier post time.

Viewers cannot fully rely on the vote scores to detect broken links, as broken links are common across posts with different vote scores. Figure 5 shows the distribution of the proportions of broken links among the links in the questions and the answers with different vote scores. These proportions range from 15.6% to 19.8% in questions and 11.1% and 13.6% in answers. This shows that although Stack Overflow viewers are less likely to vote on the broken posts, broken links are common across posts with different vote scores. For questions, one possible reason is that the questions with broken links with higher view counts can have higher vote scores than the questions without broken links with lower view counts. Similarly, one possible reason for answers is that the answers with broken links can have higher vote scores than the answers to another question without broken links. When viewers browse the posts that have been voted by other viewers, it is common for them to encounter broken links. We suggest Stack Overflow could detect the broken links and mark up the broken posts.

Figure 6 shows the proportions of broken links among the links posted by the users with different reputations. The proportions decrease from 18.6% (for the users with a reputation of less than 10) to 7.4% (for the users with a reputation higher than 10^5). More specifically, we observe that the proportions of broken links among the links posted by different users and the user reputation are significantly correlated with Pearson's correlation coefficient = -0.94 (p-value $<$ 0.05) [60]. **This shows that users with higher reputations are associated with fewer broken links, i.e., the links posted by users with a higher reputation are more likely to be permanent links.** We suggest viewers

33. https://meta.stackexchange.com/q/94826/138723#138723



(a) Question

(b) Answer

Fig. 5: The proportions of the broken posts among the posts with links based on the vote scores. These figures show that broken links are common across questions and answers with different vote scores.

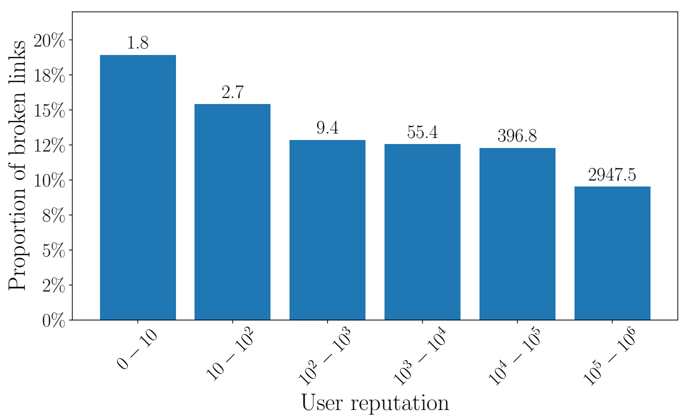


Fig. 6: The proportion of broken links among the links posted by the users with different reputations. The text label above each bar displays the number of links that are posted by the users with different reputations. These figures show that broken links are less common in users with a higher reputation.

use user reputations to detect broken links, as viewers are less likely to encounter broken links in the posts that are posted by the users with higher reputations. One possible reason is that the users with lower reputations ask only one or two questions with a limited number of links, e.g., the links to show examples. In Figure 6, the text label above each bar displays the number of links that are posted by the users with different reputations. More specifically, we observe that the proportions of broken links among the links posted by different users and the number of links posted by users with different reputations are significantly correlated with Spearman's rank correlation coefficient = -1.0 (p-value < 0.05) [58]. One link being broken leads to a large proportion of broken links among the links posted by the users with lower reputations. In contrast, the users with higher reputations usually answer questions more frequently with a larger number of links. One link being broken leads to a small proportion of broken links among the links posted by the users with higher reputations.

> Only 1.57% of the broken posts are highlighted as such by viewers in the posts' comments. Only 5.8% of the broken posts removed the broken links. For the questions with the same range of view counts, questions without broken links are associated with higher vote scores compared with the questions with broken links. For the answers to the same questions, answers without broken links are associated with higher vote scores compared with the answers with broken links. Viewers cannot fully rely on the vote scores to detect broken links, as broken links are common across posts with different vote scores. Users with a higher reputation are associated with fewer broken links.

5.3 Which Websites are Referenced by Broken Links the Most on Stack Overflow?

Motivation: It is still unclear which websites are referenced by broken links on Stack Overflow the most. By understanding the websites referenced by broken links, Stack Overflow could pay more attention to the websites that are referenced by broken links the most.

Approach: To understand the websites that are referenced by broken links on Stack Overflow the most, we perform both quantitative and qualitative analysis. In the quantitative analysis, we captured an overall picture of the websites referenced by broken links. More specifically, we first quantify the numbers of broken links that reference to different websites and analyze their distribution. For the websites referenced by different number of links on Stack Overflow, we also analyze whether the websites referenced by more links on Stack Overflow are more likely to be referenced by broken links.

We analyze the websites that are referenced by broken links the most to summarize the types of root causes of the broken links and suggest the corresponding detection and fixing strategies. We take the top 20 websites ordered by the number of broken links referencing to them as an example. The number of broken links referencing to the top 20 websites accounts for 27.6% of the broken links on Stack Overflow. Table 5 shows the top 20 websites ordered by the number of broken links referencing to them. We also present the dominant response code. To analyze the website types of the top 20 websites ordered by the number of broken links referencing to them, we manually performed a lightweight open coding process [61]. This process involves 2 phases and is performed by the first two authors (i.e., A1 and A2) of this paper:

- Phase I: The first two authors independently categorized the types of websites. For the websites that can be accessed currently, the first two authors checked the content of the websites. For the websites that cannot be visited currently, the first two authors referred to related descriptions. For example, the first two authors searched the website on Google and read the web pages that are presented in search results, e.g., the Internet Archive's Wayback Machine[34]. The first two authors took notes regarding the deficiency or ambiguity of the labeling for these websites. Table 4 presents the types of websites identified in this phase.
- Phase II: The first two authors discussed the coding results to resolve any disagreements until a consensus was reached. We did not invite others because all the disagreements were resolved during the discussion. For example, A1 considered `social.msdn.microsoft.com` as a documentation website because this website is the subdomain of the Microsoft documentation website (i.e., `msdn.microsoft.com`). A2 considered `social.msdn.microsoft.com` as a forum website because 90% of the links to this website connect to the `https://social.msdn.microsoft.com/Forums` sub-path. Finally, we consider `social.msdn.microsoft.com` as a forum website, and the resources hosted in this website can be maintained by the users and moderators. The first two authors maintained the coding schema to resolve schema deficiencies and ambiguities. No new website types were added during this discussion. The interrater agreement of this coding process has a Cohen's kappa of 0.93 (measured before discussion), indicating that the agreement level is high [53].

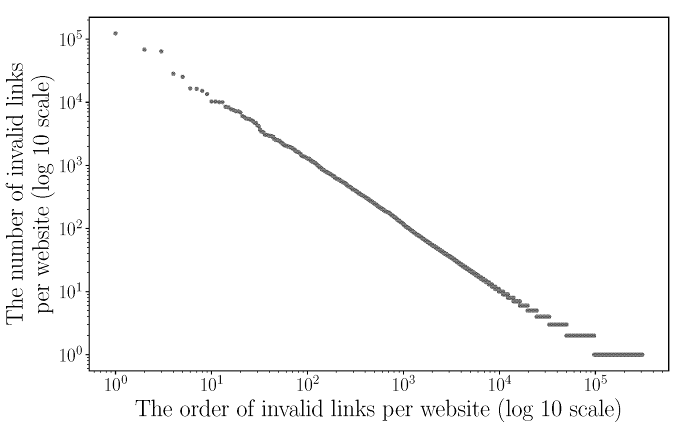


Fig. 7: The distribution of the numbers of broken links that reference to different websites in descending order. These figures show that the numbers of the broken links in different websites conform to the power-law distribution.

34. https://archive.org/web/

Results:

5.3.1 Quantitative Results

50% (i.e., 844,002) of broken links reference to the top 0.3% (i.e., 414) websites in terms of the number of the broken links referencing to them. 308,737 (i.e., 46.9%) of the websites that are referenced by Stack Overflow are referenced by broken links. Figure 7 shows the plot of the numbers of broken links that reference to different websites on Stack Overflow. The numbers of the broken links that reference to different websites conform to the power-law distribution with $\alpha = 1.96$ and $xmin = 10.0$.

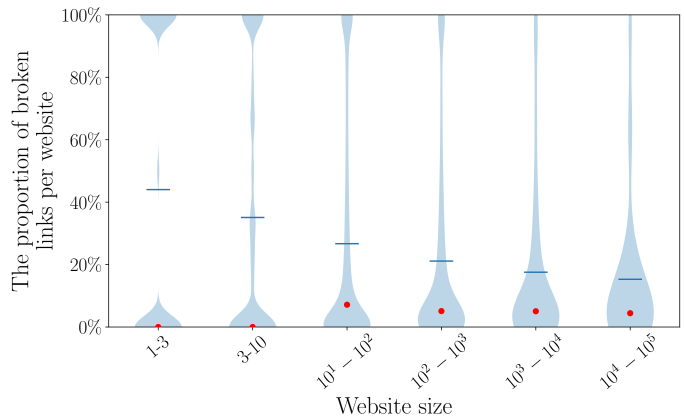


Fig. 8: The distribution of the proportions of broken links among the links that reference to websites of different appearance frequencies (i.e., the number of links reference to the website on Stack Overflow). This figure shows that websites that are referenced by fewer links on Stack Overflow are more likely to have broken links.

Figure 8 plots the proportions of broken links among the links that reference to websites of different appearance frequencies (i.e., the number of links reference to the website on Stack Overflow). We refer to websites with higher appearance frequencies to be more popular websites. To check whether the differences in the proportions of broken links among the links that reference to different websites are statistically significant between the websites with different appearance frequencies, we perform the Kruskal-Wallis H-test [62]. The null hypothesis is that there is no difference between the websites with different appearance frequencies in terms of the proportion of broken links. As a result, the differences between the websites with different appearance frequencies are significant (p-value $<$ 0.05). We then calculate Cliff's delta to measure the effect size [56]. As a result, the differences between the websites with different appearance frequencies is small in terms of the proportions of broken links among the links that reference to them (Cliff's delta is between 0.147 and 0.33). More specifically, we observe that **websites that are referenced by fewer links on Stack Overflow are more likely to be referenced by broken links.** For example, `github.com` is the second most commonly shared external website on Stack Overflow (1,870,707 links on Stack Overflow reference to `github.com`). 6.6% of the links that reference to `github.com` are broken links. The broken links that reference to `github.com` account for 7.3% of the broken links on Stack Overflow. In contrast,

TABLE 4: The website types of the top 20 websites in terms of the number of broken links referencing to them.

Type	Function	Maintainer	Example
Code	Share code projects, code snippets, and runnable code examples.	users	github.com, pastebin.com, jsfiddle.net
Documentation	Share official development related documentation of a product.	official teams	docs.oracle.com
Official	Share a starting point to other resources of the product.	official teams	www.microsoft.com
File Hosting	Provide file hosting services.	users	www.dropbox.com
Image Hosting	Provide online image sharing and hosting services.	users	i.stack.imgur.com

among 468,577 links that reference the websites with 1–3 links that are shared on Stack Overflow, 43% of them are broken links. This proportion is 6.5 times higher than the proportion of broken links among the links that reference to `github.com`. The broken links that reference the websites with 1–3 links that are shared on Stack Overflow account for 16% of the total number of broken links. This indicates that the websites with 1–3 links that are shared on Stack Overflow contribute 2.2 times broken links on Stack Overflow more than `github.com`. The more popular websites are less likely to be referenced by broken links in terms of their proportion.

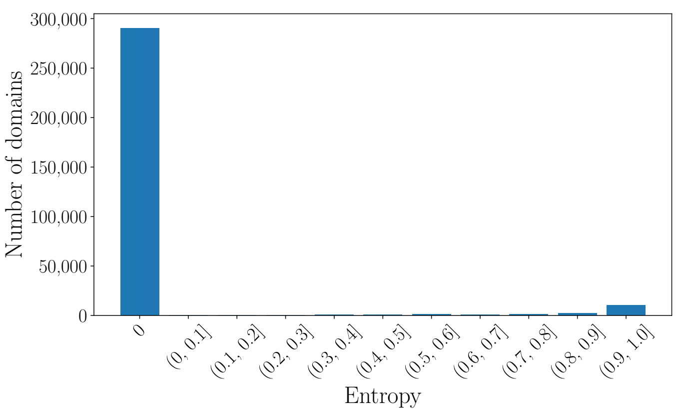


Fig. 9: The distribution of the normalized entropy of the response code of the broken links that reference to each website. This figure shows that most of the broken links that reference to each website are caused by the same reason.

Table 5 shows the dominant response code of the broken links that reference to different websites on Stack Overflow. Over 90% of the broken links that reference to 12 websites are responded with the same response code. For example, 93.15% of the broken links that reference to jsfiddle.net are responded with 404 response code. To test whether most of the broken links that reference to the same websites are responded with the same response code is common in Stack Overflow, for each website that are referenced by broken links on Stack Overflow, we calculate its the normalized entropy of the response code of the broken links. Different from a simple statistic of the total number of hits by error codes suffice, normalized entropy can describe the randomness of the information in the groups with different sizes. A non-uniform distribution would have less normalized entropy (i.e., less random) than a uniform distribution. In this paper, we use the normalized entropy to measure the randomness of the response code of the broken links in each website. We calculate the normalized entropy by dividing the entropy with the number of broken links that point to that website. Figure 9 plots the normalized entropy of the response code of the broken links that reference to each website. As a result, most of the normalized entropies are 0, indicating that **most of the broken links that reference to different website are caused by the same reason**.

5.3.2 Qualitative Results

Table 5 shows the top 20 websites ordered by appearance frequencies, as well as the dominant response code. **Among the top 20 websites ordered by appearance frequencies, 15 websites host the resources that can be maintained by their users.** For example, resources hosted in 10 *code* websites, 3 *file-hosting* websites, 1 *forum* websites, and 1 *image* websites can be maintained by their users, such as deleting the resources that can be referenced by links. 8 of the websites that host the resources that can be maintained by their users are mainly responded with a 404 response code, i.e., a client-side error and indicating that the resources hosted on that links cannot be found. This shows that one of the possible reasons for the broken links that reference to the websites that can be maintained by their users is that users can delete the resources according to their judgment. For example, the comments to a post[35] that answers how to auto-resize the input field with jQuery indicate that

> *That link leads to GitHub's 404 page.* - CoderDennis
> *Yeah, the author removed it. See my updated answer.* - Dmitry Pashkevich

This comment shows that the resources that are hosted on other websites (i.e., GitHub) can be removed by the owner of the resources. To help viewers avoid broken links when browsing Stack Overflow, we suggest viewers be cautious about the links that host resources that can be maintained by their users. To maintain the crowdsourced knowledge on Stack Overflow, we suggest that Stack Overflow should archive snapshots of links to backup the resources that are maintained by users.

Posts with links that reference to code websites are the ones that are the least maintained. For example, *github.com* are referenced by the largest number of broken links on Stack Overflow. 123,774 (i.e., 6.6%) of all broken links on Stack Overflow reference to *github.com*. The resources hosted on *github.com* is maintained by their users. 82.97% of the broken links reference to this website are responded with 404, indicating that these resources are removed from *github.com*. We suggest the Stack Overflow users not reference to the resources hosted on code websites. We encourage users to paste code within the Stack Overflow websites, e.g., using code blocks or Stack Snippets [63], [64].

Resources hosted in 4 *documentation* websites and 1 *official* website are maintained by the official teams. For example, *developer.apple.com* is referenced by broken links the most among the websites that are maintained by the

35. https://stackoverflow.com/q/9065853/

TABLE 5: The top 20 websites in terms of the number of broken links on Stack Overflow.

Website	Website Type	% among all broken links	% in website	Dominant Status Code	% Dominant Status Code
github.com	code	7.33%	6.62%	404	82.97%
codepen.io	code	4.05%	63.68%	403	97.80%
pastebin.com	code	3.81%	58.32%	403	95.96%
jsfiddle.net	code	1.50%	2.13%	404	93.15%
code.google.com	code	0.97%	8.67%	405	76.72%
developer.apple.com	documentation	0.80%	6.10%	404	93.17%
gist.github.com	code	0.61%	9.75%	404	80.84%
grepcode.com	code	0.60%	83.11%	TCPTimedOutError	62.06%
msdn.microsoft.com	documentation	0.60%	1.08%	503	49.75%
social.msdn.microsoft.com	forum	0.50%	22.75%	Error	98.16%
www.microsoft.com	official	0.49%	21.65%	404	61.49%
pastie.org	code	0.46%	91.14%	500	47.39%
dl.dropboxusercontent.com	file hosting	0.45%	90.63%	404	97.84%
dl.dropbox.com	file hosting	0.43%	90.48%	404	98.94%
postimg.org	image sharing	0.41%	94.28%	DNSLookupError	100.00%
docs.oracle.com	documentation	0.34%	1.49%	404	94.03%
docs.djangoproject.com	documentation	0.33%	6.53%	404	99.98%
drive.google.com	file hosting	0.32%	22.07%	404	96.38%
fiddle.jshell.net	code	0.32%	73.38%	404	99.78%
ideone.com	code	0.30%	7.91%	404	98.54%

official teams. This website is the least maintained documentation website among all the websites referenced on Stack Overflow. One possible reason for the broken links that reference to the developer.apple.com website is related to the deprecation of APIs. For example, a comment to an accepted answer[36] to where to get the standalone executable binary CVS for OSX indicates that

> *Sadly, as of Feb 2014, those are both dead links. According to developer.apple.com/library/ios/releasenotes/DeveloperTools/...* [37] *CVS and RCS have been removed as of Xcode 5. The new official CVS web page seems to be savannah.nongnu.org/projects/cvs* [38]

This comment shows that the webpages that host the deprecated APIs are directly removed without being archived. We suggest the designers of the websites that are maintained by the official teams could redirect the requests for the outdated web pages to an updated web page.

> 50% (i.e., 844,002) of broken links reference to the top 0.3% (i.e., 414) websites in terms of the number of the broken links referencing to them. Websites that are referenced by fewer links on Stack Overflow are more likely to be referenced by broken links. The websites that host resources maintained by their users are the least maintained, e.g., github.com.

5.4 Are the posts and comments Associated with Particular Tags More Likely to Have Broken Links than Others?

Motivation: It is still unclear the posts and comments associated with which tags suffer from the broken links the most on Stack Overflow. By understanding the severity of the broken links problem associated with different Stack Overflow tags, we could suggest that the viewers who seek solutions of the questions related to certain tags should be more cautious.

Approach: To find out which tags are the most associated with broken links, we first obtain the question threads (i.e., a Stack Overflow questions, together with its answers and comments) that reference to the broken links. Then we extract the tags of each question from the Stack Overflow data dump `Posts`. In this paper, given an answer or a comment, we use the tags of its corresponding question as its tags. Finally, we group the broken links into different tags and count the numbers.

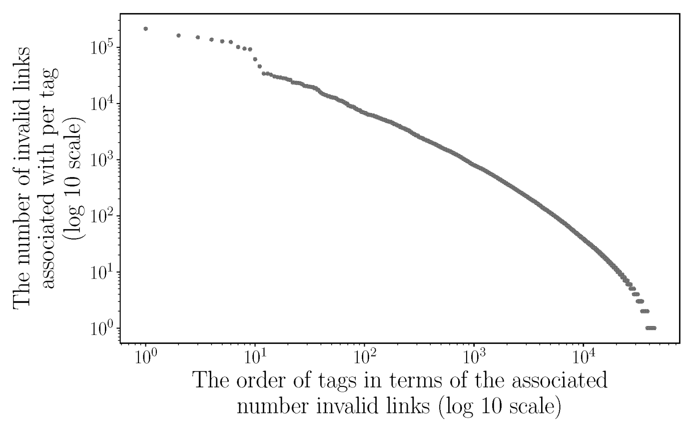


Fig. 10: The number of broken links associated with different tags in descending order.

Results: Among 55,027 tags on Stack Overflow, the posts and comments related to 54,083 tags contain links, and the posts and comments related to 44,413 (i.e., 82.1%) tags contain broken links. Figure 10 shows the plot of the numbers of broken links associated with different tags. **The top 10 tags in terms of the number of associated broken links corresponds to 55.4% of the broken links on Stack Overflow.** We suggest Stack Overflow be cautious of the

36. https://stackoverflow.com/q/1252397/
37. https://developer.apple.com/library/ios/releasenotes/DeveloperTools/RN-Xcode/
38. http://savannah.nongnu.org/projects/cvs

TABLE 6: Top 10 tags in terms of the number of broken links. This Table shows that web development related technologies are more likely to have broken links.

Tag	# Broken Links	# Links	% Broken Links
javascript	213,532	1,668,150	12.80%
php	162,454	769,853	21.10%
java	149,973	982,938	15.26%
html	137,063	1,045,135	13.11%
css	127,719	954,516	13.38%
jquery	123,606	970,833	12.73%
c#	100,880	890,799	11.32%
android	94,348	690,778	13.66%
python	92,122	773,885	11.90%
c++	61,316	472,542	12.98%

question threads that are associated with the top 10 tags in terms of the number of associated broken links.

Table 6 shows the top 10 tags in terms of the number of the associated broken links. **The posts and comments related to the web technologies, i.e., JavaScript, HTML, CSS, and jQuery, are associated with more broken links.** When Stack Overflow viewers browse the posts and comments related to the knowledge on web technologies, they are more likely to cannot fully understand the questions. We suggest that Stack Overflow should pay more attention to the posts and comments related to web technologies.

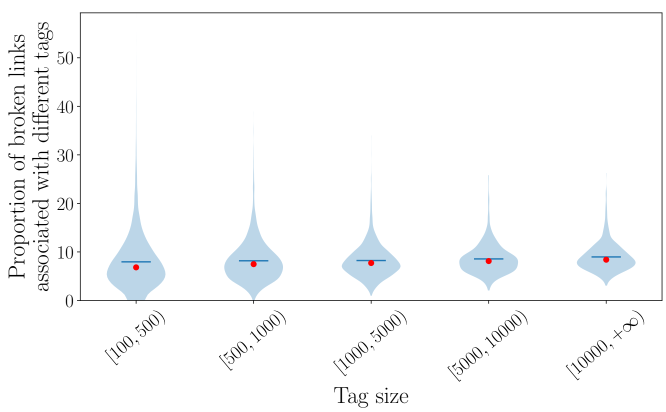


Fig. 11: The proportions of broken links among the links that are associated with the tags with different usage frequencies (i.e., the number of question threads that are associated with certain tags) This figure shows that broken links are more common for popular tags on Stack Overflow.

Figure 11 shows the violin plots of the proportions of broken links among the links that are associated with the tags with different usage frequencies (i.e., the number of question threads that are associated with certain tags). We refer to tags with higher usage frequencies to be more popular tags. To check whether the differences in the proportions of broken links among the links that are associated with different tags are statistically significant between the tags with the different usage frequencies, we perform the Kruskal-Wallis H-test [62]. The null hypothesis is that there is no difference between the tags with the different usage frequencies in terms of the proportions of broken links. As a result, the differences between the tags with the different usage frequencies are significant (p-value < 0.05). We then calculate Cliff's delta to measure the effect size [56]. As a result, the differences between the tags with the different usage frequencies are small in terms of the proportions of broken links (Cliff's delta is between 0.147 and 0.33). More specifically, we observe that the tags with 100 to 500 questions threads have the least proportion of broken links, and the tags with over 10,000 questions threads have the largest proportion of broken links. This shows that **broken links are more common for popular tags on Stack Overflow.**

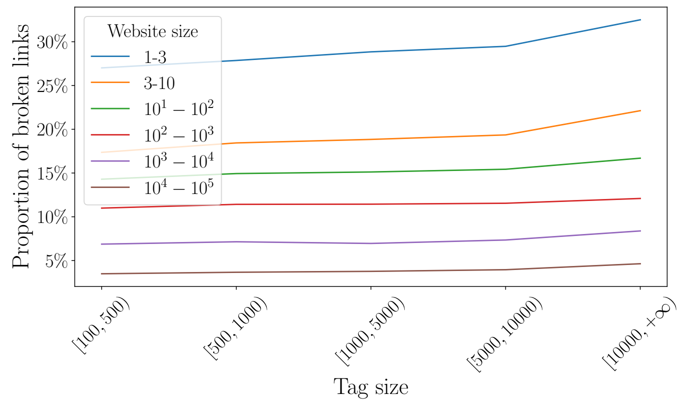


Fig. 12: For the tags with different usage frequencies, this figure plots the proportion of broken links among the links that reference to the websites with different appearance frequencies. This figure shows that the links that reference to less popular websites on Stack Overflow are more likely to be broken in the popular tags.

For the tags with different usage frequencies, Figure 12 plots the proportion of broken links among the links that reference to the websites of different appearance frequencies. We observe that the average proportions of broken links among the links that reference to the websites with 1–3, 3–10, 10^1–10^2, 10^2–10^3, 10^3–10^4, 10^4–10^5 links that are shared on Stack Overflow and the usage frequencies of tags are significantly correlated with Pearson's correlation coefficients from 0.92 (for 10^4–10^5) to 0.98 (for 10^1–10^2) (p-values < 0.05) [60]. The average proportions of broken links among the links that reference to the websites with 10^6–10^7 links that are shared on Stack Overflow and the usage frequencies of tags are not significantly correlated with Pearson's correlation coefficient = 0.81 (p-value = 0.10) [60]. For example, for the tags with 100–500 question threads, the proportion of broken links among the links that reference to the websites with 1–3 links that are shared on Stack Overflow is 39.4%. But for the tags with over 10,000 question threads, the proportion of broken links among the links that reference to the websites with 1–3 links that are shared on Stack Overflow is 46.4% (i.e., 1.18 times higher than that in the tags with 100–500 question threads). For the tags with 100–500 question threads, the proportion of broken links among the links that reference to the websites with over 10,000 links that are shared on Stack Overflow is 6.5%. But for the tags with over 10,000 question threads, the proportion of broken links among the links that reference to the websites with over 10,000 links that are shared on Stack Overflow is 6.9% (i.e., 1.07 times higher than that in the tags with 100–500 question threads). This shows that **in popular tags, the broken links are more likely to reference**

to the websites that are referenced by fewer links on Stack Overflow.

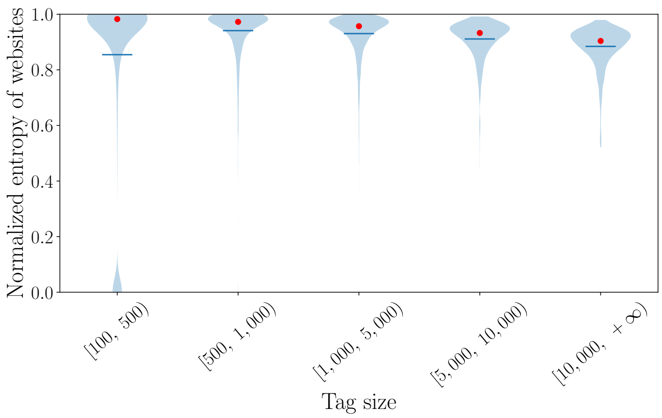


Fig. 13: The distribution of the normalized entropy of the websites that are associated with the broken links based on the tags with different usage frequencies. This figure shows that the broken links associated with most tags are uniformly referencing to different websites. The broken links in popular tags are less likely to uniformly referencing to different websites.

Figure 13 shows the distribution of the normalized entropy of the websites that are associated with each tag. More specifically, the normalized entropies of 75% tags are higher than 0.8 in terms of the websites that are associated with them. This indicates that **broken links associated with most tags are uniformly referencing to different websites.** This shows that focusing on repairing the broken links that reference to specific websites cannot help with the broken links problem associated with a certain tag. To check whether the differences between the tags with different usage frequencies are significant in the normalized entropies of the websites that are associated with them, we perform the Kruskal-Wallis H-test [62]. The null hypothesis is that there is no difference between the tags with different usage frequencies in terms of the entropies of the websites that are associated with them. As a result, the differences are significant (p-value < 0.05). We then compute Cliff's delta to measure the effect-size [56]. As a result, the differences are large (Cliff's delta > 0.474). This shows that the **broken links in popular tags are less likely to uniformly referencing to different websites**, i.e., the broken links in popular tags are more likely to centrally referencing a limited number of websites.

Figure 14 shows the distribution of the normalized entropy of the response code of the broken links that are associated with the tag with different usage frequencies. To check whether the differences between the tags with different usage frequencies are significant in the normalized entropies of the response code of the broken links that are associated with them, we perform the Kruskal-Wallis H-test [62]. The null hypothesis is that there is no difference between the tags with different usage frequencies in terms of the entropies of the response code of the broken links that are associated with them. As a result, the differences are significant (p-value < 0.05). We then compute Cliff's delta to measure the effect-size [65]. As a result, the differences

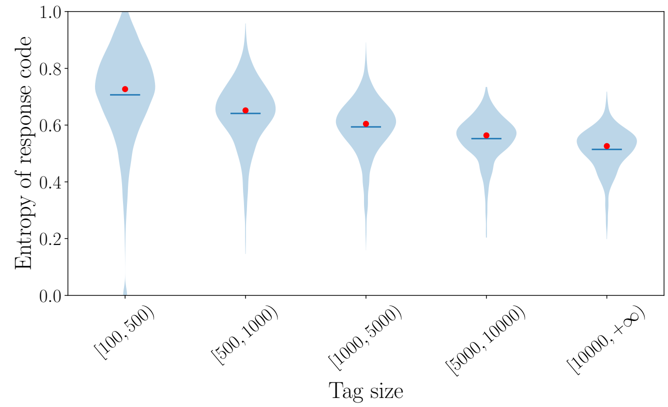


Fig. 14: The distribution of the normalized entropies of the response code of the broken links that are associated with the tags with different usage frequencies. This figure shows that the response code of the broken links that are associated with the more popular tags is less random.

are large (Cliff's delta > 0.474). **The broken links in popular tags are less likely to be associated with different response codes.** One possible reason is that the broken links in popular tags are more likely to centrally referencing a limited number of websites and most of the response code of the broken links that reference to the same websites are the same.

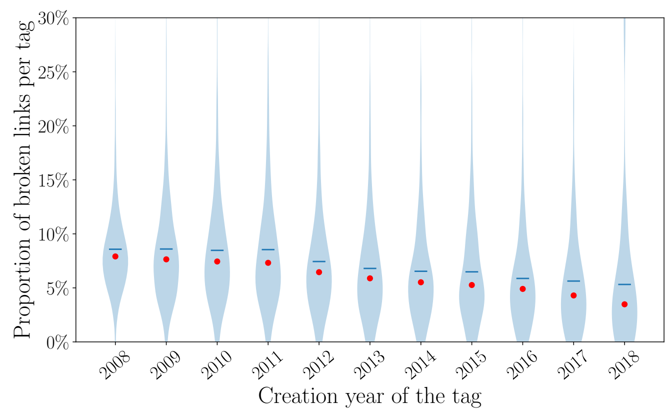


Fig. 15: The proportions of broken links among the links that are associated with the tags that were created in different years (we identify the creation of a tag as the first question that is marked with the tag were posted on Stack Overflow). This figure shows that older tags are more likely to have broken links.

Figure 15 shows the violin plots of the proportions of broken links among the links associated with different tags based on the creation time of tags (i.e., when the first question that is marked with the tag were posted on Stack Overflow). To check whether the differences in the proportions of broken links among the links associated with different tags are statistically significant between the tags with different creation dates, we perform the Kruskal-Wallis H-test [62]. The null hypothesis is that there is no difference between tags with different creation dates in terms of the proportions of broken links. As a result, the differences between the tags with different creation dates are significant (p-value < 0.05). We then compute Cliff's delta to measure the effect-size [65]. As a result, the difference between tags with different creation dates in terms of the proportions of broken links are large (Cliff's delta > 0.474). More specifically, **older tags have a larger proportion of broken links compared with the younger tags.** One possible reason is that the question threads associated with older tags still host the knowledge that is posted to solve the problems a long time ago. These nuggets of knowledge may be out-of-date, and the websites that host the out-of-date knowledge may not be maintained anymore.

> 50% of the broken links are referenced in the threads that are assigned at least one of the following 10 tags: JavaScript, PHP, Java, HTML, CSS, jQuery, C#, Android, Python, and C++. Broken links are more common for popular tags on Stack Overflow. Posts related to web technologies, e.g., JavaScript, HTML, CSS, and jQuery, are associated with more broken links. We also observe that popular tags and older tags are more likely to have broken links.

6 Discussion

In this section, we compare the difference in the vote scores between the broken posts and the normal posts in two scenarios, i.e., before the identification of broken links and after the identification of broken links. We also characterize how broken links were used on Stack Overflow and discuss the implications of our findings for Stack Overflow moderators, users, and researchers. We also consider the threats to the validity of our results.

6.1 Are the broken posts receiving fewer votes after the identification of broken links?

In Section 5.2, we compare the accumulative difference between the broken posts and the normal posts in terms of vote scores. In this section, we would like to investigate how the broken links related to the accumulation of vote scores over time. More specifically, we compare the difference in the vote scores between the broken posts and the normal posts in two scenarios, i.e., before the identification of broken links and after the identification of broken links.

To estimate when links were broken, we use the crawled data in WayBack Machine[39] to check when the links were broken for a sample of broken links. WayBack Machine is the largest digital archive of the World Wide Web with 475 billion web pages that were crawled since 1996. In terms of how links are logged, WayBack Machine uses the open-source web crawler project, Heritrix[40], to crawl websites [66]. More specifically, the Heritrix crawler would scan the web page, identify hyperlinks, collect all the linked pages, and so on [67]. Wayback Machine archives only publicly accessible pages. It does not archive the pages protected by passwords or "do not crawl" exclusions (e.g., robots.txt files that disallow access), and the pages with embedded dynamic content (e.g., as enabled by JavaScript).[41] In terms of the frequencies of collection, Wayback Machine collects material at a measured, adaptive pace. For example, Arora, Sanjay K., et al. observed that some pages (e.g., home pages or whole websites associated with highly visible organizations) may be crawled more often than other pages, and there is no readily available explanation describing the variance in capture-rates between one page (or site) and another [68].

We use the Wayback Availability JSON API[42] to obtain the timestamp of the closest available snapshot of the broken links on Stack Overflow. The closest available snapshots of the broken links are the most recently successfully archived. For the broken links that are not archived on WayBack Machine, the Wayback Availability JSON API will not return any timestamp, i.e., an empty JSON object. For the broken links that are archived on WayBack Machine, links were not broken before the timestamp of the closest available snapshot. More specifically, similar to Section 3.2, we request the Wayback Availability JSON API using Scrapy. To avoid the IP being banned from the website, we used the proxies obtained in Section 3.2 and made different requests using different proxies. We set up each proxy making one request at the same time. Finally, we obtain the timestamps of the closest available snapshots of 635,883 (i.e., 37.7%) broken links in Stack Overflow history. This shows that the Internet Archive does not archive all the broken links on Stack Overflow. We suggest Stack Overflow to archive snapshots of the links that are shared on Stack Overflow by themselves.

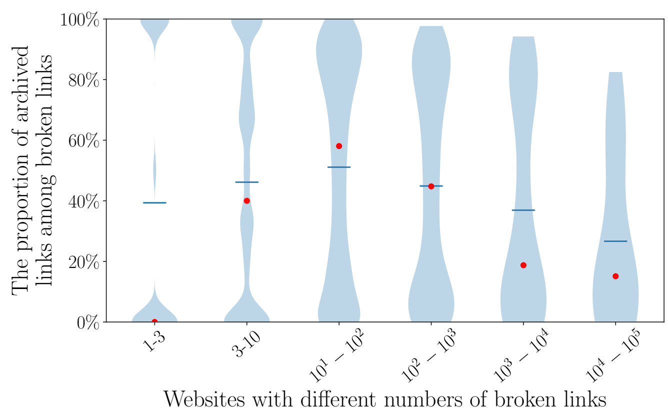


Fig. 16: The proportions of the archived links among the websites with different numbers of broken links.

To better understand the broken links that are archived by WayBack Machine, we characterize the proportion of archived links among the websites with different numbers of broken links on Stack Overflow. Figure 16 shows the proportions of archived links among the websites with different numbers of broken links on Stack Overflow. To check whether the differences in the proportions of the archived links are statistically significant between the websites with different numbers of broken links, we perform the KruskalWallis H-test [62]. The null hypothesis is that

39. http://wayback.archive.org/
40. http://www.crawler.archive.org/index.html
41. https://help.archive.org/hc/en-us/articles/360004716091-Wayback-Machine-General-Information
42. https://archive.org/help/wayback_api.php

there is no difference between the websites with different numbers of broken links in terms of the proportions of the archived links. As a result, there is no difference between the websites with different numbers of broken links in terms of the proportions of the archived links (p-value > 0.05). This shows that the archived broken links are common across the websites with different numbers of broken links on Stack Overflow. However, the websites with more broken links have a large absolute number of broken links that have not been archived. For example, for all the 11 websites with 10^4–10^5 broken links on Stack Overflow, 364,081 broken links were not archived. The non-archived broken links in 11 websites with 10^4–10^5 broken links account for 34.6% of the non-archived broken links. We suggest Stack Overflow to encourage WayBack Machine to archive the websites that host more broken links on Stack Overflow.

For the broken links that are archived by WayBack Machine, we first analyze whether the votes to the normal posts are more than the votes to the broken posts before the identification of broken links. To do so, we collect the votes to the broken posts before the timestamp of the closest available snapshot in the first year after being posted. We also collect the vote to the normal posts in the first year after being posted to see whether there is any difference between the normal posts and the broken posts before the identification of broken links. For the broken links that are archived on WayBack Machine, links were not identified as broken before the timestamp of the closest available snapshot. Figure 17a shows the numbers of votes per post per month of the normal posts and the broken posts before the timestamp of the closest available snapshot in the first year after being posted. To check whether the differences in the number of votes per post per month before the identification of broken links are statistically significant between the broken posts and the normal posts, we perform a Mann Whitney test [55]. We also calculate Cliff's delta to measure the effect size [56]. The null hypothesis is that there is no difference between the broken posts and the normal posts in terms of the number of votes per post per month before the identification of broken links. As a result, the difference between broken posts and the normal posts in the number of votes per post per month before the identification of broken links is significant and the effect size is large (p-value < 0.05 and Cliff's delta > 0.474). This indicates that **for the broken links that are archived by WayBack Machine, the votes to normal posts are less than the votes to broken posts before the identification of broken links.**

Then, for the broken links that are archived by WayBack Machine, we analyze whether the votes to the normal posts are more than the broken posts after the identification of broken links. To do so, we collect the votes to the broken posts one year before the collection of the dataset (i.e., Jun. 2, 2019) after the timestamp of the closest available snapshot. To see whether there is any difference between normal posts and broken posts after the identification of broken links, we also collect the votes of the normal posts one year before the collection of the dataset. However, Wayback Machine archive websites with varied frequency, e.g., daily, weekly, monthly, quarterly, or annual[43,44]. For the broken links that are archived on WayBack Machine, the links could be broken for a long time after the closest available snapshot, i.e., the links can be not broken in our collected data. This can lead to that the votes to the broken posts are more than the normal posts in our collected data. Figure 17b shows the numbers of votes per post per month of the broken posts and the normal posts one year before the collection of the dataset after the timestamp of the closest available snapshot. To check whether the differences between the broken posts and the normal posts are statistically significant in the number of votes per post per month after the identification of broken links, we perform a Mann Whitney test [55]. We also calculate Cliff's delta to measure the effect size [56]. The null hypothesis is that there is no difference between the broken posts and the normal posts in terms of the number of votes per post per month after the identification of broken links. As a result, the difference between broken posts and the normal posts in the number of votes per post per month after the identification of broken links is significant and the effect size is large (p-value < 0.05 and Cliff's delta > 0.474). This indicates that **for the broken links that are archived by WayBack Machine, the votes to the normal posts are more than the broken posts after the identification of broken links.**

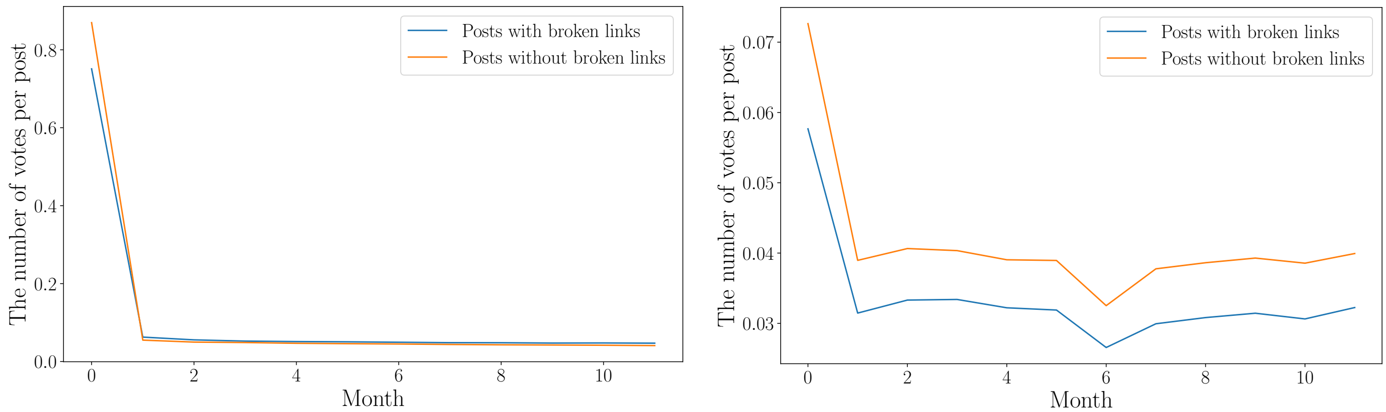
6.2 How broken links were used on Stack Overflow?

To mitigate the negative impact brought by broken links, Stack Overflow sets up community norms in their users' guides. For example, Stack Overflow suggests questioners copy the code of the live example of the problem [6]. Stack Overflow also suggests answerers quote the most relevant part of an important link [7]. However, it is still unclear how broken links were used on Stack Overflow.

To label how broken links were shared on Stack Overflow, we manually performed two lightweight open coding processes applied on the 768 posts (i.e., 384 questions and 384 answers) that are sampled from Section 5.1. More specifically, we first performed a lightweight open coding process to check the **anchor text** of the shared broken links. This process involves 3 phases and is performed by the three authors of this paper (i.e., A1, A2, and A3):

- Phase I: We randomly selected 100 broken links from the sampled 768 broken links. A1 first developed a draft coding schema (i.e., categories) of the anchor text using 100 randomly selected broken links. Then A2 and A3 used the draft coding schema to categorize the anchor text of the same 100 broken links collaboratively. During this phase, the coding schema of the anchor text of the broken links was revised and refined. At the end of Stage 1, we obtain 4 categories of the anchor text of the broken links on Stack Overflow
- Phase II: A1 and A2 applied the resulting coding schema of Phase I to categorize the remaining 668 broken links independently. They were instructed to take notes regarding the deficiency and ambiguity of the coding schema for

43. https://support.archive-it.org/hc/en-us/articles/208111686-Glossary-of-Archive-It-and-Web-Archiving-Terms
44. https://help.archive.org/hc/en-us/articles/360004651612-Archive-It-Information



(a) The numbers of votes per post per month in the year since the sharing of links or the creation of the posts before the identification of broken links.

(b) The numbers of votes per post per month one year before the collection of the dataset after the identification of broken links.

Fig. 17: The vote scores to the broken posts and the normal posts before the identification of broken links and after the identification of broken links.

categorizing certain broken links. The inter-rater agreement (Cohen's kappa) of this stage is 0.91, indicating that the agreement level is high [53].

- Phase III: A1, A2, and A3 discussed the coding results obtained in Phase 2 to resolve the disagreements. No new categories were added during this phase. At the end of Stage 3, we obtained the final coding schema and the final coding results of the sampled 768 broken links. Table 7 shows the categories of the anchor text of broken links in Stack Overflow posts.

Then we performed another lightweight open coding process to check **how the content of the broken links was posted**. We also labeled whether users followed the community norms, i.e., quoted the content of any links in the posts into quotation boxes, code blocks, or code snippets. Similarly, this process involves 3 phases and is performed by three authors of this paper (i.e., A1, A2, and A3):

- Phase I: We randomly selected 100 broken links from the sampled 768 broken links. A1 first developed a draft coding schema (i.e., categories) of how broken links were used on Stack Overflow using 100 randomly selected URL references. Then A2 and A3 used the draft coding schema to categorize the same 100 broken links collaboratively. During this phase, the coding schema of how broken links were used on Stack Overflow was revised and refined. At the end of Stage 1, we obtain 5 categories of how broken links were used on Stack Overflow.
- Phase II: A1 and A2 applied the resulting coding schema of Phase I to categorize the remaining 668 broken links independently. They were instructed to take notes regarding the deficiency and ambiguity of the coding schema for categorizing certain broken links. The inter-rater agreement (Cohen's kappa) of this stage is 0.66, indicating that the agreement level is substantial [53].
- Phase III: A1, A2, and A3 discussed the coding results obtained in Phase 2 to resolve the disagreements. For example, in a question[52] on how to center the logo on a responsive site, the questioner indicates that,

I would like the logo and the image slider to center on my responsive site.
Basically, the logo and the slider are next to each other on a full size screen. I can make the slide disappear when the site isn't wide enough and the logo shrinks.
What I want to do is make the logo and the slider center once they are no longer next to each other.
Please help: http://ranchocordovaeventscenter.com/
Thank you, Matt

A1 considered the broken link is referenced with the text description with plain URL. A2 considered the broken link is referenced solely in plain URL, as there is no code related to the link in the post. We finally considered this broken link is referenced with the text description with plain URL. The code related to the link in the post should be a description of the link that hosts the code of the problem. The first two authors maintained the coding schema to resolve schema deficiencies and ambiguities. At the end of Stage 3, we obtained the final coding schema and the final coding results of the sampled 768 broken links. Table 8 shows the categories of how the content of the link is posted in the post.

45. https://stackoverflow.com/q/25500840/
46. http://cubiq.org/iscroll-4
47. https://stackoverflow.com/q/7622772/
48. http://ideone.com/y3l7Q
49. https://stackoverflow.com/q/7853339/
50. http://www.sqlmag.com/article/sql-server/virtual-auxiliary-table-of-numbers
51. https://stackoverflow.com/q/11314236/
52. https://stackoverflow.com/q/15562201/
53. http//www.omegahat.org/RSPython/
54. https://stackoverflow.com/q/2573204/
55. http://confluence.jetbrains.net/display/ReSharper/OutOfMemoryException+Fix
56. https://stackoverflow.com/q/13652792/
57. http://sb1.collagekingapp.com/apple/iphone-4/filter/art_and_graphics.html
58. https://stackoverflow.com/q/37028082/
59. http://www.pocomatic.com/docs/whitepapers/ioc/
60. https://stackoverflow.com/q/991517/
61. http://codepad.org/Qktxh475

TABLE 7: Categories of anchor text of broken links.

Category	Definition	Example
URL	The URL is directly posted. The anchor text of the link is the URL itself.	Have a look at http://codepad.org/Qktxh475[45]
Topic	The anchor text is the topic of the shared link.	you can have a look at iscroll[46] and ... [47]
Demonstrative	The anchor text uses pronouns to refer to the Url.	See it in action.[48][49]
Author	The author name of the link is used to refer to the Url.	... technique is due to Itzik Ben-Gan[50].[51]

TABLE 8: Categories of how the content of the link is posted in the post.

Category	Definition	Example
Anchor	Only the topic of the content of the link is shared in the anchor text in the post.	At the moment, three options: RPy, RPy2, and RSPython[53].[54]
Text Topic	Only the topic of the content of the link is shared in the text blocks.	JetBrains claims this is an issue with Visual Studio process fragmentation. http://confluence.jetbrains.net/dis... [55][56]
Description	The description of the content of the link is shared in the text block.	please visit link[57]. on top left, you can see ... once you mouse-over on that, you can see ... [58]
Quotation	The shared content is quoted in the quotation box in the post.	... will be found in Why and what of Inversion of Control[59] by Ke Jin.:[60] *Besides, the imperative natural of these classic ...*
Code	The shared code is quoted in the code blocks or code snippets in the post.	Have a look at http://codepad.org/Qktxh475[61] [62] $<?\ url = "http://my-site.co.jp/user/fblogin$ ...
No	No content related to the link is ever mentioned in the post.	when going through this guide: https://confluence.atlassian.com/display/... [63] and copying the script, it fails to achieve the desired result.[64]

TABLE 9: Distribution of how broken links were shared on Stack Overflow.

	Topic	URL	Demonstrative	Author	
Question					
Anchor	15%	0%	0%	0%	15%
Text topic	3%	12%	3%	0%	17%
Description	1%	7%	1%	0%	8%
Quotation	0%	0%	0%	0%	1%
Code	1%	6%	5%	0%	12%
No	1%	40%	6%	0%	47%
	20%	65%	15%	0%	100%
Answer					
Anchor	35%	0%	0%	0%	35%
Text topic	1%	9%	5%	0%	15%
Description	2%	4%	1%	0%	8%
Quotation	1%	3%	1%	0%	5%
Code	1%	4%	4%	1%	10%
No	1%	18%	7%	1%	26%
	42%	39%	17%	2%	100%

Table 9 summarizes the distribution of broken links in different anchor text forms and summarized forms. We observe that **40% of the sampled questions directly post the URL of the broken link without any content related to the link in the post (i.e., URL only link).** Note that the "URL only link" does not indicate that the post only has a URL without other content. The question description of the example in Table 8 contains the background of the question, and the description of the failure, etc. However, readers cannot obtain any further valuable information from the URL only links. For example, a reader with a similar problem cannot gain much information from the link to the tutorial. The "URL only link" can be related to that in Section 5.2, we observe that questioners are the users with lower reputations that only ask one or two questions. They may not know how to ask a question well.

62. https://stackoverflow.com/q/25500840/
63. https://confluence.atlassian.com/display/BITBUCKET/Set+up+SSH+for+Git
64. https://stackoverflow.com/q/29304997/

35% of the sampled answers only summarize the content of the shared broken links as the anchor text of the URL (i.e., anchor topic link),. These answers do not quote or describe the content of the link. Readers cannot directly obtain the content hosted on the link. When the links are broken, readers have to use the content topic of the link to search for the knowledge related to the link. One possible reason for the anchor topic link is that too many quotation boxes in the answers can confuse readers from the parts that are the most useful to them. We suggest Stack Overflow could learn from Wikipedia[65], where article previews are popped up in a small window when readers hover the cursor over links.

In terms of whether users follow the community norms, we observe that 21.1% (i.e., 81) of the sampled answers and 15.1% (i.e., 58) of the sampled questions with broken links quote the content of any links in the posts into quotation boxes, code blocks, or code snippets. This shows that **users commonly not follow the Stack Overflow community norms.** 70.1% (i.e., 60) answers and 84.5% (i.e., 49) questions that follow the community norms quote the broken links. This shows that users would quote the broken links when they need to quote an important link. We suggest researchers design a tool to identify the links that need quotation.

6.3 Implications

6.3.1 Actionable Suggestions for Researchers

Implication 1: Future work could repair the broken links on Stack Overflow based on the revisions of links. In Section 5.2, we observe that broken links have negative impacts on the crowdsourced knowledge on Stack Overflow. We also observe that 2,458,323 revisions repaired the broken links. For example, the broken link `http://dev.mysql.com/doc/refman/5.5/en/reserved-words.html` is replaced by a valid link `http://dev.mysql.com/doc/refman/5.5/en/keywords.html` in 7 posts.

65. https://en.wikipedia.org/wiki/Wikipedia:Tools/Navigation_popups

However, there are still 949 posts that reference this broken link. A more general case is that the linked site changes the location of the resource. For example, 384 posts share the links that reference to `ant.apache.org` website with the path `manual/CoreTasks` in history. All links (i.e., 63) that reference to `ant.apache.org` website with the path `manual/CoreTasks` are broken links (i.e., the server responded with 404 response code). 152 posts replace the path `manual/CoreTasks` with `manual/Tasks`. This is because the website change the location of the documentation related to *Tasks*.[66] We suggest that the future researchers could repair the broken links on Stack Overflow based on the revisions of links.

6.3.2 Actionable Suggestions for Stack Overflow

Implication 2: To help viewers avoid broken links when browsing Stack Overflow, Stack Overflow should detect the broken links and mark the questions with broken links. In Section 5.2, we observe that viewers vote more on the normal posts than the broken posts after the links become broken. This shows the negative impact of the broken links. We encourage Stack Overflow to take action on these broken links. More specifically, Stack Overflow could learn from Wikipedia[67] where links can be reviewed, replaced with a working or archive link, tagged, or removed. On Stack Overflow, moderators could use a tool, e.g., W3C checklink[68], and Xenu's Link Sleuth[69], to automatically scan links to identify the broken links. After that, Stack Overflow could include an *broken link* mark for the questions with broken links. By doing so, Stack Overflow users can be aware of the broken links when browsing the Stack Overflow websites even before clicking them.

Implication 3: To maintain the crowdsourced knowledge on Stack Overflow, Stack Overflow should develop mechanisms to encourage users (especially posts owners) to pay more attention to the broken links and make efforts to maintain any broken links. In Section 6.3.1, we observe that only 1.7% of the broken posts are notified by the viewers in comments and only 5.8% of the broken links that are posted on the Stack Overflow history are removed. After notified by the viewers in comments, only 14.3% of the posts ever with broken links repair the broken links. This shows that the Stack Overflow gamification system (i.e., vote and accept answers) fails to encourage users to comment out and update broken links on Stack Overflow. We suggest the Stack Overflow moderators could adjust the gamification system to encourage users to identify and update broken links. For example, Stack Overflow could reward badges or reputation scores to users who identify or maintain broken links.

Implication 4: Stack Overflow could archive snapshots of links as soon as they were created. In Section 5.3, we observe that the links that reference to the websites that host the resources that can be maintained by users are the least maintained. These resources can be customized created, maintained, and deleted by their users. More specifically, Stack Overflow could learn from Google[70] and the Internet Archive's Wayback Machine[71] to archive snapshots of links when the links are posted. Google takes a snapshot of each web page as a backup in case the current page is not available. Internet Archive provides free universal access to books, movies, music, as well as 458 billion archived web pages. However, these external web archive services cannot capture all the content of all the links at the time when the links are posted. For example, a comment to the accepted answer that provides AI tools/frameworks/Library for Objective C[72] complains that:

> *The A* link appears to be dead. The WayBack machine captured part of the post, although it missed most (only has DemoView.m and a small blurb remain). web.archive.org/web/20090207003416/http://bravobug.com/news/...* [73]

This comment shows that the external web archive service only captures part of the content[74]. Stack Overflow could periodically replace the broken links on Stack Overflow with the links to the copies of the resources in the archive.

6.3.3 Actionable Suggestions for Users

Implication 5: Stack Overflow users are encouraged not to remove the examples in the links in Stack Overflow questions. In Section 5.1, we observe that 65% of the broken links in our sampled questions are used to show examples, e.g., code examples. This shows that the examples hosted in the links in the questions is removed after the problems are resolved. However, this practice would lead to these questions with broken links to be totally useless as no following viewers can understand the questions. We suggest Stack Overflow users not remove the examples of the links, especially in questions.

Implication 6: We recommend that Stack Overflow users post the code in a more permanent site, e.g., code blocks and Stack Snippets on Stack Overflow as much as possible, rather than the ephemeral external code websites, e.g., github.com. In Section 5.3, we observe that code websites host the largest number of broken links. This shows that the links that reference on code websites on Stack Overflow are often ephemeral (they get broken after some time has passed). Stack Overflow provides code blocks for users to paste code snippets, and even Stack Snippets enable users to post runnable code [63], [64]. We strongly recommend Stack Overflow users to post the code in the code blocks and Stack Snippets as much as possible, rather than external code websites.

6.3.4 Feedback from Stack Overflow

To understand whether our research can characterize broken links problems and obtain useful findings for Stack Overflow, we shared our findings with Stack Overflow community[75]. They concurred with our findings and see the

66. https://github.com/apache/ant/commit/61b4c00b3852083b0f81-586d6f78adf0bc3c7f6f
67. https://en.wikipedia.org/wiki/User:Dispenser/Checklinks
68. http://validator.w3.org/checklink
69. http://home.snafu.de/tilman/xenulink.html
70. https://support.google.com/websearch/answer/1687222
71. https://archive.org/web/
72. https://stackoverflow.com/q/5533317/
73. http://web.archive.org/web/20090207003416/http://bravobug.com/news/?p=118
74. http://bravobug.com/news/?p=118
75. https://meta.stackexchange.com/q/353998/

importance of broken links problems on Stack Overflow (*Six comments complain about the broken links. That is six missed opportunities to fix those links ...* – rene, *I really admire your time and dedication, great effort.* – Shadow Wizard Wearing Mask). Motivated by that some of the respondents asked for the practical implications and solutions (*Interesting findings, but what are practical implications and solutions? Why was this endeavor carried out? Your post could use some introduction.* – Luuklag), we also conducted a new survey to validate claims of the usefulness of the study in Section 6.3.5. However, respondents could directly see the values of our findings and implications (*"We suggest SO should directly archive the links when links are posted." In addition to keeping links alive, this also solves the problem of all that reviewer time wasted on deleting link-only answers. Win!* – francescalus). They were interested in our work and requested for more details. For example, they requested us to present our research related to the proportion of broken links among the links that were posted each month. Based on our findings, the Stack Overflow community was also interested in investigating the soft 404 problems (*Many links redirect to some generic page (may or may not be covered by response codes) or show content like "Content could not be found" or similar. In other words, many sites try to hide the fact that a link is broken. For instance, if the returned page is very short it could be counted as a broken link (though some may contain a link to a new location). Or some commonly used phrases could be detected.*). We suggest future research efforts should continue working with the Stack Overflow team to solve/alleviate the broken links problem.

6.3.5 Feedback from Stack Overflow users

To validate the usefulness of the research findings, we conduct a survey by spreading a questionnaire to a broad range of companies from various locations worldwide.[76] We received 84 responses, and 64 (i.e., 76.2%) of the responses are valid responses (i.e., the respondents use Stack Overflow during their development process). Table 10 shows the feedback of our research findings from Stack Overflow users.[77] Appendix **??** presents more details about how we conduct the survey.

We observe that 70.3% to 89.1% of the respondents agree or strongly agree with the implications in our paper. For example, a respondent comments that searching for information related to broken links on search engines is time-consuming. They need a tool that can automatically repair broken links or recommend web pages related to broken links. Another respondent believes that marking the posts with broken links would save time for them when browsing on Stack Overflow. Two respondent comments that they would maintain links if they can gain more badges or reputation scores.

Some respondents show disagreements about our implications and comment on their concerns. For example, a respondent refuses to use Stack Snippet because he thinks that Stack Snippet is not as good as GitHub. We suggest Stack Overflow could survey their users to understand what Stack Overflow can learn from GitHub in terms of code pasting. However, 76.6% of the respondents agree or strongly agree that Stack Overflow users should post the code in a permanent website, code block, or Stack Snippet on Stack Overflow. For example, a respondent considers that the code is the most important part of a Stack Overflow answer. He considers that a post with a code broken link would have a negative impact.

Another respondent disagrees with the implication that Stack Overflow should archive the snapshots of links when links are shared. He is concerned about the copyright of the archived webpages. We suggest Stack Overflow should follow the policies related to the copyright if they decide to archive the snapshots of links. However, 70.3% of the respondents agree or strongly agree that Stack Overflow should archive the snapshots of links when links are shared. A respondent comments that he would consider the archive of the links as a quick-fix of the broken links.

The results of our survey highlight the usefulness of the research findings and implications of the paper.

6.4 Threats to Validity

Threats to internal validity concern the factors that could have influenced our results. We heavily depend on manual processes as described in Section 5.1 and Section 5.3. Like any human activity, our manual labeling process is subject to personal bias. To reduce personal bias in the manual labeling process, each website was labeled by two of the authors and discrepancies were discussed until a consensus was reached. We also showed that the level of inter-rater agreement of the qualitative studies is high (i.e., the values of Cohen's kappa ranged are between 0.69 to 0.96).

In Section 4.3, we observe that 87,086 broken links were posted by the URL Rewriter Bot in May 2017. The URL Rewriter Bot is used by Stack Overflow to automatically update the schema of the links for security and privacy concerns without checking their validity. When replacing HTTP with HTTPS in URL, the URL Rewriter Bot might replace a valid link with a broken link. However, only 3.5% of the broken links are posted by URL Rewriter Bot. This indicates that broken links being posted by URL Rewriter Bot brings limited threats to our work.

In Section 5.2, we analyze the difference of accumulative popularity (e.g., vote scores) between broken posts and normal posts. We observe that broken links are common across the posts with different accumulative vote scores. However, a link can be broken at any time. To compare the difference in the vote scores between the broken posts and the normal posts in two scenarios, i.e., before the identification of broken links and after the identification of broken links, in Section 6.1 we use the crawled data in WayBack Machine to check when the links were broken for a sample of broken links. Prior studies observe that Wayback Machine collects material at an adaptive pace [68]. There could be a threat that the link can not be archived by Wayback Machine if the site is not visited for a long-time. Our finding shows that for the broken links that are archived by WayBack Machine, the votes to normal posts are less than the votes to broken posts before the identification of broken links, and

76. Our questionnaire can be found in https://forms.gle/YDqd9fMN7KG8mMs78 (English version) and https://www.wjx.cn/vm/wCBvsCU.aspx (Chinese version).

77. Comments in the responses to the survey are publicly available at https://zenodo.org/record/4683732.

TABLE 10: Feedback of our research findings from Stack Overflow users

	Implication 1	Implication 2	Implication 3	Implication 4	Implication 5	Implication 6
Agree or strongly agree	81.30%	82.80%	89.10%	70.30%	78.10%	76.60%
Neutral	14.20%	9.40%	7.80%	7.80%	12.50%	15.60%
Disagree or strongly disagree	4.50%	7.80%	3.10%	21.90%	9.40%	7.80%

the votes to the normal posts are more than the broken posts after the identification of broken links.

Threats to external validity concern the generalization of our findings. Our study is conducted to investigate the broken links on Stack Overflow. That said, our findings may not be generalized to the broken links in other Q&A sites. For example, other Q&A forums that focus on a particular technology, e.g., Google Product Forums[78] and Microsoft Community[79], only share the links that relate to the specific technology. In contrast, Stack Overflow is a popular website for developers and covers a wide range of programming-related technologies, and the links are prevalently shared across technologies. In the future, we plan to analyze broken links in other Q&A systems.

To validate the usefulness of our findings with more participants, we conducted an anonymous online survey with Stack Overflow users. As we asked respondents to disseminate our survey to their colleagues, our surveyed respondents may not fully represent the whole developer population. To mitigate this threat, we recruit the respondents who use Stack Overflow in their work in the industry from diverse organizations, e.g., Alibaba, ByteDance, Tencent, Microsoft, Google, Line, and other companies to collect feedback from diverse backgrounds. Furthermore, our studied population is similar to the ones previously studied in the literature [69], [70].

7 Conclusion

In this paper, we investigate the broken links on Stack Overflow. As a result, we observe that there are 1,687,995 broken links (14.2%) in the latest version of Stack Overflow posts. 65% of the broken links in our sampled questions are used to show examples, e.g., code examples. 70% of the broken links in our sampled answers are used to provide supporting information, e.g., explaining a certain concept. Only 1.57% of the broken posts are highlighted as such by viewers in the posts' comments. Only 5.8% of the broken posts removed the broken links. Viewers cannot fully rely on the vote scores to detect broken links, as broken links are common across posts with different vote scores. The websites that host resources that can be maintained by their users are referenced by broken links the most on Stack Overflow, e.g., github.com. Web technology related questions, e.g., JavaScript, HTML, CSS, and jQuery, are more likely to have broken links.

In the future, we plan to design a tool to repair the broken links on Stack Overflow based on the revisions of links. For example, if the linked site changes the location of the resource, we would like to use the new location of the resource to update the links with the old location.

78. https://productforums.google.com/forum/
79. https://answers.microsoft.com/en-us/

Acknowledgement

This research was partially supported by the National Science Foundation of China (No. U20A20173), Key Research and Development Program of Zhejiang Province (No.2021C01014), and the National Research Foundation, Singapore under its Industry Alignment Fund – Pre-positioning (IAF-PP) Funding Initiative. Any opinions, findings, and conclusions, or recommendations expressed in this material are those of the author(s) and do not reflect the views of the National Research Foundation, Singapore.

References

[1] M. M. Rahman, S. Yeasmin, and C. K. Roy, "Towards a context-aware ide-based meta search engine for recommendation about programming errors and exceptions," in *2014 Software Evolution Week-IEEE Conference on Software Maintenance, Reengineering, and Reverse Engineering (CSMR-WCRE)*. IEEE, 2014, pp. 194–203.

[2] X. Xia, L. Bao, D. Lo, P. S. Kochhar, A. E. Hassan, and Z. Xing, "What do developers search for on the web?" *Empirical Software Engineering*, vol. 22, no. 6, pp. 3149–3185, 2017.

[3] How to reference material written by others. https://stackoverflow.com/help/referencing.

[4] How do i format my posts using markdown or html? https://stackoverflow.com/help/formatting.

[5] H. Zhang, S. Wang, T. P. Chen, Y. Zou, and A. E. Hassan, "An empirical study of obsolete answers on stack overflow," *IEEE Transactions on Software Engineering*, 2019.

[6] How do i ask a good question? https://stackoverflow.com/help/how-to-ask.

[7] How do i write a good answer? https://stackoverflow.com/help/how-to-answer.

[8] R. K. Saha, A. K. Saha, and D. E. Perry, "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*, ser. ESEC/FSE 2013. New York, NY, USA: ACM, 2013, pp. 663–666.

[9] M. Linares-Vásquez, G. Bavota, M. Di Penta, R. Oliveto, and D. Poshyvanyk, "How do api changes trigger stack overflow discussions? a study on the android sdk," in *Proceedings of the 22Nd International Conference on Program Comprehension*, ser. ICPC 2014. New York, NY, USA: ACM, 2014, pp. 83–94.

[10] H. Zhang, S. Wang, T. Chen, and A. E. Hassan, "Reading answers on stack overflow: Not enough!" *IEEE Transactions on Software Engineering*, pp. 1–1, 2019.

[11] What are tags, and how should i use them? https://stackoverflow.com/help/tagging.

[12] 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*, vol. 19, no. 3, pp. 619–654, Jun 2014.

[13] C. Rosen and E. Shihab, "What are mobile developers asking about? a large scale study using stack overflow," *Empirical Software Engineering*, vol. 21, no. 3, pp. 1192–1223, Jun 2016.

[14] K. Bajaj, K. Pattabiraman, and A. Mesbah, "Mining questions asked by web developers," in *Proceedings of the 11th Working Conference on Mining Software Repositories*, ser. MSR 2014. New York, NY, USA: ACM, 2014, pp. 112–121.

[15] S. Wang, D. Lo, B. Vasilescu, and A. Serebrenik, "Entagrec: An enhanced tag recommendation system for software information sites," in *2014 IEEE International Conference on Software Maintenance and Evolution*. IEEE, 2014, pp. 291–300.

[16] B. Xu, D. Ye, Z. Xing, X. Xia, G. Chen, and S. Li, "Predicting semantically linkable knowledge in developer online forums via convolutional neural network," in *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering*, ser. ASE 2016. New York, NY, USA: ACM, 2016, pp. 51–62. [Online]. Available: http://doi.acm.org/10.1145/2970276.2970357

[17] X. Xia, D. Lo, X. Wang, and B. Zhou, "Tag recommendation in software information sites," pp. 287–296, 2013.

[18] Why can people edit my posts? how does editing work? https://stackoverflow.com/help/editing.

[19] F. Chen and S. Kim, "Crowd debugging," in *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*. ACM, 2015, pp. 320–332.

[20] L. Cai, H. Wang, Q. Huang, X. Xia, Z. Xing, and D. Lo, "Biker: a tool for bi-information source based api method recommendation," in *Proceedings of the 2019 27th ACM Joint Meeting - European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, M. Dumas, D. Pfahl, S. Apel, and A. Russo, Eds. United States of America: Association for Computing Machinery (ACM), 2019, pp. 1075–1079.

[21] Q. Huang, X. Xia, Z. Xing, D. Lo, and X. Wang, "Api method recommendation without worrying about the task-api knowledge gap," in *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*, ser. ASE 2018. New York, NY, USA: ACM, 2018, pp. 293–304. [Online]. Available: http://doi.acm.org/10.1145/3238147.3238191

[22] What does it mean when an answer is "accepted"? https://stackoverflow.com/help/accepted-answer.

[23] Vote up. https://stackoverflow.com/help/privileges/vote-up.

[24] Why is voting important? https://stackoverflow.com/help/why-vote.

[25] L. Mamykina, B. Manoim, M. Mittal, G. Hripcsak, and B. Hartmann, "Design lessons from the fastest q&a site in the west," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '11. New York, NY, USA: ACM, 2011, pp. 2857–2866.

[26] H. Cavusoglu, Z. Li, and K.-W. Huang, "Can gamification motivate voluntary contributions?: The case of stackoverflow q&a community," in *Proceedings of the 18th ACM Conference Companion on Computer Supported Cooperative Work & Social Computing*, ser. CSCW'15 Companion. New York, NY, USA: ACM, 2015, pp. 171–174. [Online]. Available: http://doi.acm.org/10.1145/2685553.2698999

[27] A. Anderson, D. Huttenlocher, J. Kleinberg, and J. Leskovec, "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*, ser. KDD '12. New York, NY, USA: ACM, 2012, pp. 850–858.

[28] A. Pal, S. Chang, and J. A. Konstan, "Evolution of experts in question answering communities," in *sixth international AAAI conference on weblogs and social media*, 2012.

[29] B. V. Hanrahan, G. Convertino, and L. Nelson, "Modeling problem difficulty and expertise in stackoverflow," in *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work Companion*, ser. CSCW '12. New York, NY, USA: ACM, 2012, pp. 91–94. [Online]. Available: http://doi.acm.org/10.1145/2141512.2141550

[30] G. Li, H. Zhu, T. Lu, X. Ding, and N. Gu, "Is it good to be like wikipedia?: Exploring the trade-offs of introducing collaborative editing model to q&a sites," *conference on computer supported cooperative work*, pp. 1080–1091, 2015.

[31] S. Wang, T.-H. P. Chen, and A. E. Hassan, "How do users revise answers on technical q&a websites? a case study on stack overflow," *IEEE Transactions on Software Engineering*, 2018.

[32] C. Chen, Z. Xing, and Y. Liu, "By the community & for the community: a deep learning approach to assist collaborative editing in q&a sites," *ACM Proceedings on Human-Computer Interaction*, vol. 1, no. CSCW, 11 2017.

[33] C. Chen, X. Chen, J. Sun, Z. Xing, and G. Li, "Data-driven proactive policy assurance of post quality in community q&a sites," *Proc. ACM Hum.-Comput. Interact.*, vol. 2, no. CSCW, pp. 33:1–33:22, Nov. 2018. [Online]. Available: http://doi.acm.org/10.1145/3274302

[34] D. Ye, Z. Xing, and N. Kapre, "The structure and dynamics of knowledge network in domain-specific q&a sites: a case study of stack overflow," *Empirical Software Engineering*, vol. 22, no. 1, pp. 375–406, Feb 2017.

[35] C. Gómez, B. Cleary, and L. Singer, "A study of innovation diffusion through link sharing on stack overflow," in *Proceedings of the 10th Working Conference on Mining Software Repositories*. IEEE Press, 2013.

[36] S. Baltes, C. Treude, and M. P. Robillard, "Contextual documentation referencing on stack overflow," *IEEE Transactions on Software Engineering*, 2020.

[37] D. Correa and A. Sureka, "Integrating issue tracking systems with community-based question and answering websites," in *2013 22nd Australian Software Engineering Conference*. IEEE, 2013, pp. 88–96.

[38] T. Wang, G. Yin, H. Wang, C. Yang, and P. Zou, "Automatic knowledge sharing across communities: a case study on android issue tracker and stack overflow," in *2015 IEEE Symposium on Service-Oriented System Engineering*. IEEE, 2015, pp. 107–116.

[39] M. Rath, J. Rendall, J. L. Guo, J. Cleland-Huang, and P. Mäder, "Traceability in the wild: automatically augmenting incomplete trace links," in *2018 IEEE/ACM 40th International Conference on Software Engineering (ICSE)*. IEEE, 2018.

[40] S. Gao, Z. Xing, Y. Ma, D. Ye, and S. Lin, "Enhancing knowledge sharing in stack overflow via automatic external web resources linking," in *2017 22nd International Conference on Engineering of Complex Computer Systems*, 2017, pp. 90–99.

[41] R. Fielding and J. Reschke, "Hypertext transfer protocol (http/1.1): Semantics and content," 2014.

[42] P. Habibzadeh, "Decay of references to web sites in articles published in general medical journals: mainstream vs small journals," *Applied clinical informatics*, vol. 4, no. 04, pp. 455–464, 2013.

[43] D. Fetterly, M. Manasse, M. Najork, and J. L. Wiener, "A large-scale study of the evolution of web pages," *Software: Practice and Experience*, vol. 34, no. 2, pp. 213–237, 2004.

[44] W. Koehler *et al.*, "A longitudinal study of web pages continued: a consideration of document persistence," *Information Research*, vol. 9, no. 2, pp. 9–2, 2004.

[45] F. McCown, S. Chan, M. L. Nelson, and J. Bollen, "The availability and persistence of web references in d-lib magazine," *arXiv preprint cs/0511077*, 2005.

[46] J. Hennessey and S. X. Ge, "A cross disciplinary study of link decay and the effectiveness of mitigation techniques," in *BMC bioinformatics*, vol. 14, no. 14. BioMed Central, 2013, p. S5.

[47] M. Klein, H. Van de Sompel, R. Sanderson, H. Shankar, L. Balakireva, K. Zhou, and R. Tobin, "Scholarly context not found: one in five articles suffers from reference rot," *PloS one*, vol. 9, no. 12, 2014.

[48] T. Zeng, A. Shema, and D. E. Acuna, "Dead science: Most resources linked in biomedical articles disappear in eight years," in *International Conference on Information*. Springer, 2019, pp. 170–176.

[49] Bullet list for similar questions on page not found. https://meta.stackexchange.com/q/231869/.

[50] S. Baltes, C. Treude, and S. Diehl, "Sotorrent: Studying the origin, evolution, and usage of stack overflow code snippets," pp. 191–194, 2018.

[51] S. Baltes, L. Dumani, C. Treude, and S. Diehl, "Sotorrent: reconstructing and analyzing the evolution of stack overflow posts," in *Proceedings of the 15th International Conference on Mining Software Repositories, MSR 2018, Gothenburg, Sweden, May 28-29, 2018*, 2018, pp. 319–330.

[52] Hypertext transfer protocol (http/1.1): Semantics and content. https://tools.ietf.org/html/rfc7231.

[53] A. J. Viera and J. M. Garrett, "Understanding interobserver agreement: The kappa statistic," *Family Medicine*, vol. 37, no. 5, pp. 360–363, 2005.

[54] comment everywhere. https://stackoverflow.com/help/privileges/comment.

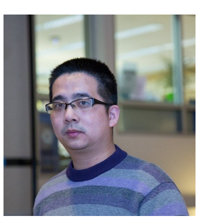
[55] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," *The annals of mathematical statistics*, pp. 50–60, 1947.

[56] N. Cliff, *Ordinal methods for behavioral data analysis*. Psychology Press, 2014.

[57] J. Romano, J. D. Kromrey, J. Coraggio, J. Skowronek, and L. Devine, "Exploring methods for evaluating group differences on the nsse and other surveys: Are the t-test and cohen'sd indices the most appropriate choices," in *annual meeting of the Southern Association for Institutional Research*. Citeseer, 2006, pp. 1–51.

[58] E. C. Fieller, H. O. Hartley, and E. S. Pearson, “Tests for rank correlation coefficients. i,” *Biometrika*, vol. 44, no. 3/4, pp. 470–481, 1957.
[59] I. Weir, “Spearman’s correlation. statstutor,” *Mathematics Education Centre Loughborough University Available at: http://www statstutor ac uk/resources/uploaded/spearmans pdf Accessed*, vol. 17, 2017.
[60] J. Benesty, J. Chen, Y. Huang, and I. Cohen, “Pearson correlation coefficient,” in *Noise reduction in speech processing*. Springer, 2009, pp. 1–4.
[61] C. B. Seaman, “Qualitative methods in empirical studies of software engineering,” *IEEE Trans. Softw. Eng.*, vol. 25, no. 4, p. 557–572, Jul. 1999. [Online]. Available: https://doi.org/10.1109/32.799955
[62] W. H. Kruskal and W. A. Wallis, “Use of ranks in one-criterion variance analysis,” *Journal of the American statistical Association*, vol. 47, no. 260, pp. 583–621, 1952.
[63] Markdown help: Code and preformatted text. https://stackoverflow.com/editing-help/#code.
[64] I’ve been told to create a “runnable” example with “stack snippets”, how do i do that? https://meta.stackoverflow.com/q/358992/.
[65] N. Cliff, “Dominance statistics: Ordinal analyses to answer ordinal questions.” *Psychological bulletin*, vol. 114, no. 3, p. 494, 1993.
[66] L. Price, “Internet archiving-the wayback machine,” 2011.
[67] A. Brown, *Archiving websites: a practical guide for information management professionals*. facet publishing, 2006.
[68] S. K. Arora, Y. Li, J. Youtie, and P. Shapira, “Using the wayback machine to mine websites in the social sciences: a methodological resource,” *Journal of the Association for Information Science and Technology*, vol. 67, no. 8, pp. 1904–1915, 2016.
[69] K. A. Safwan and F. Servant, “Decomposing the rationale of code commits: the software developer’s perspective,” in *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2019, pp. 397–408.
[70] B. Xu, Z. Xing, X. Xia, and D. Lo, “Answerbot: Automated generation of answer summary to developers’ technical questions,” in *2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2017, pp. 706–716.

Jiakun Liu Jiakun Liu is currently a Ph.D. student in the College of Computer Science and Technology, Zhejiang University, China. His research interests include mining software repositories and empirical software engineering.



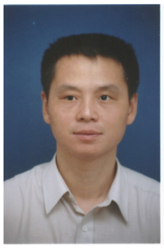
Xin Xia Xin Xia is the director of the software engineering application technology lab, Huawei, China. Prior to joining Huawei, he was an ARC DECRA Fellow and a lecturer at Monash University, Australia. Xin received his Ph.D in computer science from Zhejiang University in 2014. To help developers and testers improve their productivity, his current research focuses on mining and analyzing rich data in software repositories to uncover interesting and actionable information. More information at: https://xin-xia.github.io/

David Lo David Lo is a ACM Distinguished Member and a Professor of Computer Science at Singapore Management University. He received his PhD in Computer Science from National University of Singapore in 2008. His research interest is in the intersection of software engineering and data science, encompassing socio-technical aspects and analysis of different kinds of software artefacts, with the goal of improving software quality and developer productivity. His work has been published in premier and major conferences and journals in the area of software engineering, AI, and cybersecurity.

Haoxiang Zhang Haoxiang Zhang is a Senior Researcher at the Centre for Software Excellence at Huawei, Canada. His research interests include empirical software engineering, mining software repositories, and intelligent software analytics. He received a PhD in Computer Science from Queen’s University, Canada. He received a PhD in Physics and MSc in Electrical Engineering from Lehigh University, and obtained his BSc in Physics from the University of Science and Technology of China. Contact haoxiang.zhang@huawei.com. More information at: https://haoxianghz.github.io/.

Ying Zou Ying Zou is the Canada Research Chair in Software Evolution. She is a professor in the Department of Electrical and Computer Engineering, and cross-appointed to the School of Computing at Queen’s University in Canada. She is a visiting scientist of IBM Centers for Advanced Studies, IBM Canada. Her research interests include software engineering, software evolution, software analytics, and service-oriented architecture. More about Ying and her work is available online at https://www.ece.queensu.ca/people/Y-Zou/

Ahmed E. Hassan Ahmed E. Hassan is an IEEE Fellow, an ACM SIGSOFT Influential Educator, an IEEE TCSE Distinguished Educator, an NSERC Steacie Fellow, the Canada Research Chair (CRC) in Software Analytics, and the NSERC/BlackBerry Software Engineering Chair at the School of Computing at Queen’s University, Canada. His research interests include mining software repositories, empirical software engineering, load testing, and log mining. He received a PhD in Computer Science from the University of Waterloo. He spearheaded the creation of the Mining Software Repositories (MSR) conference and its research community. He also serves/d on the editorial boards of IEEE Transactions on Software Engineering, Springer Journal of Empirical Software Engineering, and PeerJ Computer Science. Contact ahmed@cs.queensu.ca. More information at: http://sail.cs.queensu.ca/



Shanping Li Shanping Li received his Ph.D. degree from the College of Computer Science and Technology, Zhejiang University in 1993. He is currently a professor in the College of Computer Science and Technology, Zhejiang University. His research interests include Software Engineering, Distributed Computing, and the Linux Operating System.

Appendix A
Survey Stack Overflow users

We conducted an anonymous online survey with Stack Overflow users to validate the usefulness of our findings with more participants.

A.1 Survey design

We follow Kitchenham and Pfleeger's guidelines for personal opinion surveys [71]. To filter out the respondents who may not understand our survey (i.e., respondents never used Stack Overflow), we collected demographic information about the respondents. More specifically, we asked the following question in questionnaire:

- Which options can describe your relationship with Stack Overflow?

We provided five options, including (1) "I have no relationship with Stack Overflow", (2) "I use Stack Overflow", (3) "I work for Stack Overflow", (4) "I perform scientific research using Stack Overflow"and (4) other. Based on the selections of respondents, we could exclude invalid responses.

Then respondents were explicitly asked to answer each question with respect to their experience with Stack Overflow. Respondents were expected to score our implication (i.e., Implication 1–6) related to broken links on Stack Overflow according to the "Agreement Level" (Strong Agree, Agree, Neutral, Disagree, and Strong Disagree) (i.e., Likert scales). The respondents can also provide comments and rationale supporting their selections. To ensure that respondents have a basic understanding of the characteristics of the broken links on Stack Overflow, we present the findings of our paper and the related implications. To reduce the possibility of respondents providing arbitrary answers, the respondent has the option (i.e., "I don't know") to specify that she prefers not to answer or does not understand the description of a particular question.

To support respondents from China, we translated our questionnaire to Chinese. The reason is that English is an international lingua franca, and Chinese is the most spoken language. We expected that a large number of our survey recipients are fluent in one of these two languages. We chose to make our survey available both in English on Google Forms, and in Chinese on a popular survey website in China.[80] We carefully translated our questionnaire to make sure there exists no ambiguity between English and Chinese terms in our survey.

We followed Dillman's recommended three-stage process to pre-test the survey [72]. First, the questionnaire was reviewed by colleagues and experts in the field of software engineering to uncover potential misunderstandings or unexpected outcomes. Next, we discussed the questionnaire's clarity and motivation with developers that use Stack Overflow during their development process. Finally, we piloted the preliminary survey with 14 Stack Overflow users. Note that all the respondents during the pre-test process of the survey were not our survey takers. We obtained feedback on (1) whether the length of the questionnaire was appropriate, and (2) the clarity and understandability of the terms. We made minor modifications to the preliminary questionnaire based on the received feedback and produced a final version. Note that the collected responses from the pilot survey are excluded from the presented results in this paper.

80. https://www.wjx.cn/

A.2 Respondents recruitment

To recruit respondents from Stack Overflow users, we spread the survey to a broad range of companies from various locations worldwide. To increase the response rates, we conduct the survey anonymously [73]. We asked developers from different IT companies, e.g., Alibaba, ByteDance, Tencent, Microsoft, Google, Line, and other companies to collect feedback from diverse backgrounds. We also used snowball sampling for our survey, asking respondents to disseminate our survey to their colleagues. By doing so, we could recruit the respondents that using Stack Overflow in their work in the industry from diverse organizations.

A.3 Data analysis

We received a total of 84 responses, and further excluded 20 responses made by respondents who claimed that they have no relationship with Stack Overflow. In the end, we had a set of 64 valid responses and no respondent claimed that they have other types of relationship with Stack Overflow. Our studied population was similar to the ones previously studied in the literature [69], [70]. The top two countries where the respondents reside are China (48) and United States (8). The respondents have an average of 2.1 years of professional experience (min: 0.5, max: 6). Our survey respondents are distributed across different groups (e.g., job roles). More specifically, among 60 respondents who have reported their occupations, 51.7% of them are industrial or freelance professionals, 11.7% of them are academic or industrial researchers, and 36.7% of them are undergraduate/-graduate students. We analyzed the survey results based on the question types. For Likert-scale questions, we reported the percentage of each rating is selected. Table 10 shows the feedback of our research findings from Stack Overflow users. We also analyzed comments (i.e., D11) and described some of them in Section 6.3.5. Comments in the survey are publicly available at https://zenodo.org/record/4683732.