# Characterizing Duplicate Code Snippets between Stack Overflow and Tutorials

Manziba Akanda Nishi
*Department of Computer Science*
*Virginia Commonwealth University*
Richmond, Virginia
nishima@vcu.edu

Agnieszka Ciborowska
*Department of Computer Science*
*Virginia Commonwealth University*
Richmond, Virginia
ciborowskaa@vcu.edu

Kostadin Damevski
*Department of Computer Science*
*Virginia Commonwealth University*
Richmond, Virginia
kdamevski@vcu.edu

***Abstract*—Developers are usually unaware of the quality and lineage of information available on popular Web resources, leading to potential maintenance problems and license violations when reusing code snippets from these resources. In this paper, we study the duplication of code snippets between two popular sources of software development information: the Stack Overflow Q&A site and software development tutorials. Our goals are to empirically understand the scale of repeated information between these two sources, to gain insight into why developers copy information from one source to the other, and to understand the evolution of duplicated information over time. To this end, we correlate a set of nearly 600 tutorials on Android available on the Web to the SOTorrent dataset, which isolates code snippets from Stack Overflow posts and tracks their changes over time. Our findings reveal that there are over 1,400 duplicate code snippets related to Android on Stack Overflow. Code that was duplicated on the two sources is effective at answering Stack Overflow questions; a significant number (31%) of answers that contained a duplicate code block were chosen as the accepted answer. Qualitative analysis reveals that developers commonly use Stack Overflow to ask clarifying questions about code they reused from tutorials, and copy code snippets from tutorials to provide answers to questions.**

***Index Terms*—duplicate code snippets, Stack Overflow, tutorials**

## I. Introduction

Nowadays, developers frequently consult online resources to learn new skills, expand or refresh their knowledge, or avoid repetitive tasks [1]. Code snippets (or code blocks) available on many sources of software development related information on the Web are easy to reuse and incorporate into existing projects. While reusing online code snippets improves the speed of development, it has possible negative side-effects on code quality and maintainability [2], [3]. For instance, reusing online code snippets is susceptible to introducing bugs and software vulnerabilities [4], [5] and exposing security risks as a result of outdated or poorly written code [6], [7]. Additionally, by copying and pasting code snippets with unknown origin, unaware developers can cause license violations [8], [9].

One of the largest and most visited sources of reusable code snippets is Stack Overflow, a Q&A website with a large and active community of 9.9 million users, and a corpus of 17 million questions and 26 million answers.[1] Each Stack Overflow question pertains to a specific technical problem, and may contain one or more answers that often include code blocks implementing a solution to the problem. While Stack Overflow provides answers to a large set of development problems and has a permissive license permitting reuse of posted code snippets by developers in their projects, studies show that a significant amount of posted code does not originate on this platform, but is reproduced from elsewhere [9]–[11].

Different from Stack Overflow, online tutorials provide step-by-step instructions on a specific development topic often introducing a practical application as a running example and including numerous code snippets accompanied by a rich and detailed description [12], [13]. Code snippets on tutorials are usually longer, and often several code snippets form a logical sequence interspersed with natural language explanations [14]. No tutorial source with the scale of Stack Overflow exists, and each small scale source uses its own licensing scheme.

In this study, we analyze the duplication of code snippets between tutorials and Stack Overflow, with the goal of 1) characterizing developer rationale behind reproducing snippets from source to source; and 2) understanding the scale and properties of duplicate snippets, including their evolution over time. Our findings lead towards better understanding of this phenomenon and how the two sources can be best engineered to clearly display the origin of snippets available for reuse by developers, and to improve the design of tools that mine code snippets.

## II. Research Methodology & Experimental Setup

In order to obtain easily discernible code snippets and their edit histories, we leverage the *SOTorrent* dataset, which provides the version histories for over 40 million posts, based on the official Stack Overflow data dumps, including 122 million text block versions and 77 million code snippet versions [15]. Stack Overflow data is distributed with the *Creative Commons Attribute - ShareAlike 3.0 Unported (CCBY-SA 3.0)* license. This indicates that developers must reference the original post when reusing code and must use a similar license for any derivative work.

Separately, we curated a list of 599 Android tutorials available on 5 popular software development related websites.[1] Some of the tutorial sites provided explicit licenses, while

[1] Data as of 21 January, 2019.

2574-3864/19/$31.00 ©2019 IEEE
DOI 10.1109/MSR.2019.00048

TABLE I
CURATED SET OF ANDROID TUTORIALS (AS OF 21 JANUARY, 2019).

| Tutorial Source | Number of Tutorials | Number of Java Code Snippets | Code Snippet License |
|---|---|---|---|
| vogella.com | 70 | 626 | Eclipse Public License 2.0 |
| stacktips.com | 154 | 296 | restrictive; non-standard language (all copying disallowed) |
| androidtutorialpoint.com | 82 | 622 | licensing unclear |
| tutorialspoint.com | 82 | 525 | restrictive; non-standard language (only learning use permitted) |
| sitepoint.com | 211 | 435 | licensing rights remain with post creator |
| *Total* | *599* | *2,504* | |

others used non-standard language or provided no specific license, as shown in Table I. We extracted all the code snippets from the Android tutorials based on a set of HTML tags, obtained by manually examining the patterns used to display snippets in each individual site. The HTML tags specifically focused on extracting Java code, ignoring other code blocks commonly present in Android tutorials, e.g., written in XML. As this approach failed to filter all non-Java code snippets, we used regular expressions and manual analysis to ensure that only Java code snippets remained. Following this filtering step, the tutorial corpus consisted of a total of 2,504 code snippets, ranging from 1 to 626 lines of code, and a median of 19 lines of code. Unlike the SOTorrent dataset, the edit histories of tutorials were not possible to reconstruct and, while some tutorials displayed dates of last modification, we found several examples where the dates were not updated and therefore unreliable.

In order to extract code snippets from Stack Overflow, we executed SQL queries on the BigQuery [16] interface to the SOTorrent dataset (2018-12-09 version) [15]. We filtered posts based on the android Stack Overflow tag, obtaining 3,114,844 code snippets (min = 1 LOC; max = 1,090 LOC; median = 9 LOC) with a creation date ranging from January, 2008 to January, 2018. As we already filtered the XML snippets in the tutorials, it was unnecessary to perform that step for the Stack Overflow snippet corpus.

Considering the large corpora of code snippets recovered from Stack Overflow and tutorials, in order to detect all the duplicate code segment pairs, we applied a scalable code clone detection tool, able to rapidly process a large corpora of code [17]. For scalability, the code clone detection tool uses a textual representation of source code. We used a similarity threshold value of 0.8, i.e., detecting all the code clones that have at least 80% similar terms. The threshold value of 0.8 was used in similar studies in the past [3], as it retains the flexibility of detecting Type-1, Type-2 and Type-3 code clones while producing few false positives [18] [19]. As a means to further reduce possible false positives that could occur with small code snippets, we disregarded clone pairs where one of the snippets had fewer than 10 lines of code. Following this step, we observed 4,718 duplicate code pairs between the tutorials and Stack Overflow code snippets.

In examining the detected duplicate Android code snippets, we observed a high occurrence of clone pairs that represented standard Android generated (i.e. template) code. Clearly, these snippets were not copied from either source, but rather represented well-known patterns that the Android Studio IDE generates for a few common Android classes, e.g., Activity, Fragment. In order to filter the spurious code clones, we selected a cut-off point of a maximum of 3 tutorial snippets that a single Stack Overflow snippet can map to as we observed that the larger number of tutorial matches was usually produced by code templates. We used the coarse grained filtering as a guide, and examined the dataset manually to further detect template snippets to exclude. Our final set of duplicate code pairs between tutorials and Stack Overflow consists of 2,148 duplicate pairs, representing 346 unique tutorial snippets, and 1,488 unique Stack Overflow code snippets extracted from the SOTorrent dataset.

Developers copy code snippets due to various reasons. To identify the commonly occurring justifications for code reuse between software development tutorials and Stack Overflow posts, two of the authors independently analyzed 100 randomly selected posts from Stack Overflow, including 50 questions and 50 answers, that were classified as code clones originating from tutorials. Based on the examined posts, each author devised a list of reasons (or categories) explaining why developers copied a code snippet and assigned one of them to each code clone. The agreement of authors' annotations was 92% (46 questions and 46 answers). Differences between annotation schemes were resolved via in-person discussion.

Due to the lack of reliable modification timestamps of the tutorials, we could not automatically discern which source contained the original code snippet and which source contained a copy. In many of our findings, where the source is unknown to us, we report on duplicate snippets. However, during the qualitative analysis of why snippets are copied, we were able to use contextual clues to relatively reliably predict where the snippet originated from. Such clues included links or references, existence of textual description or additional code on one of the platforms, or notions of date or time.

## III. RESEARCH FINDINGS

In this section, we present the results of our analysis of duplicate code snippets between Stack Overflow posts and Android tutorials. We first describe justifications for copying code snippets followed by an analysis of a few properties of the identified duplicates. Finally, we conclude the section specifying threats to validity of the study.

### A. Understanding Code Snippets Copied from Tutorials to Stack Overflow

We used qualitative study techniques to build a taxonomy of categories describing posts with copied code snippets following the procedure described in Section II. Based on the analysis of contextual information of the copied code blocks, the authors determined that *all of the randomly sampled*

TABLE II
CATEGORIES OF POST CONTAINING CODE SNIPPETS COPIED FROM TUTORIALS TO STACK OVERFLOW.

| Post category | Description | # Posts | Example Post |
|---|---|---|---|
| *Questions* | | | |
| Error/Exception | Facing exceptions or errors in the code | 28 / 50 | *I'm trying to put data to my list view [...] using navigation drawer. I created a list view and defined the adapter but when I run it I got null pointer in the logcat. [...] [code snippet]* |
| Unexpected behavior | Looking for help due to unexpected behavior | 13 / 50 | *I have App1 and App2. App1 has the database in the content provider and App2 will insert data in database of App1, but when I call getContentResolver().insert(...), it always return null as uri. [code snippet] Please let me know the mistake, so I can solve it.* |
| Functionality | Asking about implementation of a specific functionality | 7 / 50 | *The problem is when I rotate the cellphone, the music starts again, how can I prevent that? [...][code snippet]* |
| Version compatibility | Asking for help as a code snippet is not working with a particular version of Android API | 2 / 50 | *I want to install a library to use PreferenceFragmentCompat or any class that replaces android.app.PreferenceFragment so my app can work in API 11 and lower. Can anyone please give me some details such as which library should I use and how to install it in my AS project? [code snippet]* |
| *Answers* | | | |
| Example/Solution | Providing a solution/example implementing requested functionality | 48 / 50 | *You need to override onSaveInstanceState(Bundle savedInstanceState) and write the application state values you want to change to the Bundle parameter like this [code snippet]* |
| Fixing the code | Fixing errors in the code after developer modified code form tutorial | 1 / 50 | *Here is the fixed setup, next time you need to do the imports for each object: [fixed code snippet]* |
| Clarification | Providing additional information to support explanation | 1 / 50 | [Question:] *Why should I use an additional layout file to present a ListView?* [Answer:] *When you are creating a simple ListView: [code snippet]. When creating a custom ListView: [code snippet]. In this example, if you look at the line: View rowView = inflater.inflate(R.layout.rowlayout, parent, false);, the R.layout.rowlayout is your custom layout used to show your custom ListView. Refer to the source link at the top of the answer for a detailed tutorial on ListView's.* |

*snippets in the qualitative study were copied from tutorials to Stack Overflow*. We were unable to observe any snippets copied from Stack Overflow to tutorials, which could have been due to the specific parameters we used, e.g. minimum of 10 lines of code for a duplicate snippet. The final set of categories, identified by two of the authors, is presented in Table II. Note that depending on whether a Stack Overflow question or answer was examined, different non-overlapping sets of reasons for code reuse were discovered, hence we used questions and answers as the primary dimension in displaying the results.

The majority of cases when tutorial code snippets are copied to Stack Overflow's questions are related to experiencing an error or exception (28 out of 50 questions), or when the code does not work as intended by a developer (13 out of 50 questions). This result may be a consequence of many factors, such as e.g. errors in tutorials or a misconfigured IDE, although while analyzing questions' descriptions, we often noted that developers tried to first modify the code from tutorials and once failed, they were reaching to the Stack Overflow's community asking for help and some clarification. Copying code snippets for the purpose of presenting current implementation of a specific functionality (7 cases) occurs when a developer wants to extend the code found in a tutorial, but has little knowledge on how to proceed. Additionally, we observed 2 questions that arose due to API compatibility issues between different Android API versions.

Providing an exemplary implementation for a specific issues was a prevalent justification for reusing code snippets from tutorials in Stack Overflow's answers. In 48 out of 50 answers we observed that developers copied the code to either directly resolve the question or to present a minimal working example. Additionally, we noted a singular case of an answer fixing the code provided in the question, where the code originated from the tutorial, indicating an unsuccessful attempt of modification. Finally, we also observed one answer, categorized as clarification, when developer used the copied code to provide an example supporting explanation to a posed question.

### B. Properties and Evolution of Copied Code Snippets

We detected 2,148 duplicate snippets (346 snippets from tutorials and 1,488 snippets from Stack Overflow ) originating from 189 tutorials and 1,398 Stack Overflow posts, including 909 questions and 489 answers.

To evaluate the popularity of the reused code snippets, we studied the distribution of the number of up and down votes for Stack Overflow posts. Among 1,398 posts containing a duplicate code segment, we found 637 up voted and 226 down voted posts. Note that a post can be both up and down voted on Stack Overflow. Figure 1 shows the distribution of up and down votes with respect to the number of posts. The majority of the up voted posts received between 1 and 4 votes, while about 64 posts gathered more than 5 up votes, including one post with over 2000 up votes. Similar distribution shape is observed for the down votes, with a peak for the number of votes between 1 and 4, however less than 10 posts were down voted more than 5 times. The relatively high number of up votes and the difference between the number of up voted posts when compared to the number of down votes indicates that the code blocks copied from tutorials were considered as useful by the Stack Overflow's community. Moreover, we observed

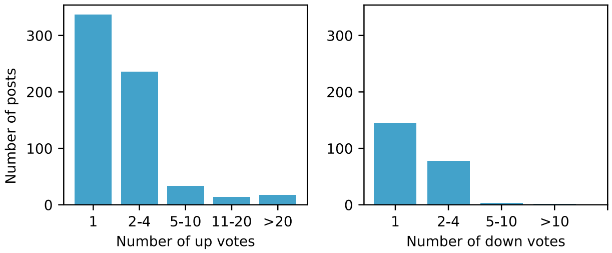

Fig. 1. Popularity distribution of Stack Overflow posts containing code clones

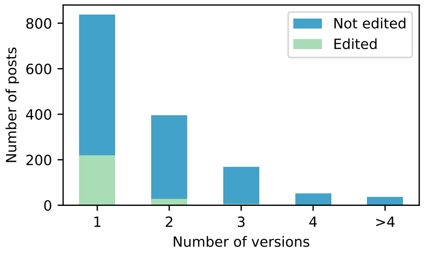

Fig. 2. The number of versions of posts containing copied code snippets

TABLE III
TIMESPAN OF EDITS FOR THE COPIED CODE SEGMENTS

| | Same year | After 1 year | After 2 years | After 3 years and more |
|---|---|---|---|---|
| First edits | 230 | 6 | 8 | 3 |
| Second edits | 33 | 1 | 4 | 3 |

that 31% of answers containing a copied code snippet were accepted as solutions to a question.

We used the SOTorrent dataset of Stack Overflow post versions to examine the edit trends of 1,488 duplicate code blocks. Note that a code block might not be edited in all versions of a Stack Overflow post, hence we distinguish between a case when two versions of a code block are the same (not-edited) or when they actually differ (edited). As a point of reference, we used an evolution analysis of Stack Overflow post blocks, including text and code blocks, presented by Baltes et al. [20]. Figure 2 shows the distribution of the code duplicates' versions, considering both not-edited and edited cases. Among all the unique duplicate code snippets, 650 (43%) have more than one version, although only 256 of them (17%) have actually been modified. Most of the edited code clones (86%) were modified once and only 1% were modified more than three times. Baltes et al. [20] reported a similar result, with 46.6% edited post blocks. Although they did not provide separate analysis for the number of edited code blocks, they observed on average 4.1 code blocks versions, whereas majority of duplicate code snippets were edited only once. This may indicate that the reused code segments are less likely to be updated as they originated from a trustworthy source, such as a tutorial.

To quantify the characteristics of modifications, we measured the difference between the number of LOC and the number of characters comparing the first and the last version of a code snippet. We observed that on average 17.4 LOC (min = 1, max = 215, median = 5.5, std = 32.1) and 545.8 characters (min = 1, max = 8596, median = 145.5, std = 1119.1) in a code block were modified (either added or deleted).

We analyzed the timespan between edits of the reused code blocks with respect to the first and second time of the modification. The results are presented in Table III. Overall, the edits characteristics of the copied code snippets follows general trend observed for Stack Overflow posts as noted in [20], with the first and second post edits occurring the same year a post was created with probability of 90.3% and 88.3% respectively. In the case of the copied code snippets, majority of the first edits (93%) take place in the same year as the post creation, while changing the code snippets the next year or later is rare, with respectively 2% and 5% of all cases. Similarly, the second edits occur most often during the year of posting the code (81%), and are less likely to be performed in the following year (2%) or later (17%). No copied code block was edited after five years of the creation. These results indicate that developers tend to edit the reused code quickly, within a short period of time of posting the code snippets.

### C. Threats to validity

Detection of the code clones is potentially susceptible to several threats. One internal threat is related to configuration of the code clone detection tool and the heuristic used to filter false positives, as it directly affects the information we use for further analysis. To mitigate this threat, we followed similar studies to configure the tool properly and examined the false positives manually to find the most suited approach to remove them. Another threat arises from the fact that we did not explicitly check for duplication of code snippets within the tutorials or Stack Overflow posts, thus these may affect the total number of detected code clones. The results of the qualitative study pose an external threat since the observations were concluded over a limited number of reused code blocks.

## IV. CONCLUSION

This paper reports on the quantity, type and evolution of duplicated source code snippets on Stack Overflow and Android tutorials. Our findings identify a set of categories describing posts containing reused code blocks and reveal some of the likely justifications for copying code from tutorials to Stack Overflow, the predominant direction of copying we encountered. Developers that reproduce the code snippets from software tutorials do so to ask queries related to observed errors or unexpected outputs or behavior. Developers sometimes use the code blocks from tutorials to provide answers to Stack Overflow questions. The answers are marked as accepted on Stack Overflow with significant ratio (31%). Our findings also reveal that the duplicated code snippets between Stack Overflow and software tutorials can evolve over time, usually within the first year of the initial post creation.

# References

[1] C. S. Corley, F. Lois, and S. Quezada, "Web usage patterns of developers," in *2015 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, Sep. 2015, pp. 381–390.

[2] J. Brandt, P. J. Guo, J. Lewenstein, M. Dontcheva, and S. R. Klemmer, "Two studies of opportunistic programming: interleaving web foraging, learning, and writing code," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2009, pp. 1589–1598.

[3] D. Yang, P. Martins, V. Saini, and C. Lopes, "Stack overflow in github: any snippets there?" in *Mining Software Repositories (MSR), 2017 IEEE/ACM 14th International Conference on*. IEEE, 2017, pp. 280–290.

[4] R. Abdalkareem, E. Shihab, and J. Rilling, "On code reuse from stackoverflow: An exploratory study on android apps," *Information and Software Technology*, vol. 88, pp. 148–158, 2017.

[5] E. Shihab, A. E. Hassan, B. Adams, and Z. M. Jiang, "An industrial study on the risk of software changes," in *Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering*. ACM, 2012, p. 62.

[6] F. Fischer, K. Böttinger, H. Xiao, C. Stransky, Y. Acar, M. Backes, and S. Fahl, "Stack overflow considered harmful? the impact of copy&paste on android application security," in *Security and Privacy (SP), 2017 IEEE Symposium on*. IEEE, 2017, pp. 121–136.

[7] Y. Acar, M. Backes, S. Fahl, D. Kim, M. L. Mazurek, and C. Stransky, "You get where you're looking for: The impact of information sources on code security," in *2016 IEEE Symposium on Security and Privacy (SP)*, May 2016, pp. 289–305.

[8] S. Baltes, R. Kiefer, and S. Diehl, "Attribution required: Stack overflow code snippets in github projects," in *Proceedings of the 39th International Conference on Software Engineering Companion*. IEEE Press, 2017, pp. 161–163.

[9] L. An, O. Mlouki, F. Khomh, and G. Antoniol, "Stack overflow: a code laundering platform?" in *Software Analysis, Evolution and Reengineering (SANER), 2017 IEEE 24th International Conference on*. IEEE, 2017, pp. 283–293.

[10] C. Ragkhitwetsagul, J. Krinke, M. Paixao, G. Bianco, and R. Oliveto, "Toxic code snippets on stack overflow," *arXiv preprint arXiv:1806.07659*, 2018.

[11] C. Ragkhitwetsagul, J. Krinke, and R. Oliveto, "Awareness and experience of developers to outdated and license-violating code on stack overflow: An online survey," *arXiv preprint arXiv:1806.08149*, 2018.

[12] L. Ponzanelli, G. Bavota, A. Mocci, M. Di Penta, R. Oliveto, M. Hasan, B. Russo, S. Haiduc, and M. Lanza, "Too long; didn't watch!: Extracting relevant fragments from software development video tutorials," in *Proceedings of the 38th International Conference on Software Engineering*, ser. ICSE '16. New York, NY, USA: ACM, 2016, pp. 261–272. [Online]. Available: http://doi.acm.org/10.1145/2884781.2884824

[13] R. Tiarks and W. Maalej, "How does a typical tutorial for mobile development look like?" in *Proceedings of the 11th Working Conference on Mining Software Repositories*, ser. MSR 2014. New York, NY, USA: ACM, 2014, pp. 272–281. [Online]. Available: http://doi.acm.org/10.1145/2597073.2597106

[14] P. Chatterjee, M. A. Nishi, K. Damevski, V. Augustine, L. Pollock, and N. A. Kraft, "What information about code snippets is available in different software-related documents? an exploratory study," in *Software Analysis, Evolution and Reengineering (SANER), 2017 IEEE 24th International Conference on*. IEEE, 2017, pp. 382–386.

[15] S. Baltes, C. Treude, and S. Diehl, "Sotorrent: Studying the origin, evolution, and usage of stack overflow code snippets," in *Proceedings of the 16th International Conference on Mining Software Repositories (MSR 2019)*, 2019.

[16] "Bigquery," https://cloud.google.com/bigquery/, 2018.

[17] M. A. Nishi and K. Damevski, "Scalable code clone detection and search based on adaptive prefix filtering," *Journal of Systems and Software*, vol. 137, pp. 130–142, 2018.

[18] J. Svajlenko, J. F. Islam, I. Keivanloo, C. K. Roy, and M. M. Mia, "Towards a big data curated benchmark of inter-project code clones," in *2014 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 2014, pp. 476–480.

[19] H. Sajnani, V. Saini, J. Svajlenko, C. K. Roy, and C. V. Lopes, "Sourcerercc: Scaling code clone detection to big-code," in *2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE)*, May 2016, pp. 1157–1168.

[20] S. Baltes, L. Dumani, C. Treude, and S. Diehl, "The evolution of stack overflow posts: Reconstruction and analysis," *arXiv preprint arXiv:1811.00804*, 2018.