# Identifying Versions of Libraries used in Stack Overflow Code Snippets

Ahmed Zerouali
ahmed.zerouali@vub.be
Vrije Universiteit Brussel
Brussels, Belgium

Camilo Velázquez-Rodríguez
camilo.ernesto.velazquez.rodriguez@vub.be
Vrije Universiteit Brussel
Brussels, Belgium

Coen De Roover
coen.de.roover@vub.be
Vrije Universiteit Brussel
Brussels, Belgium

***Abstract*—Stack Overflow is a popular question and answer platform where developers share technical issues in the hope of receiving answers with potential solutions. The latter may include code snippets making use of library versions that have long since been succeeded by newer ones. Other developers finding such a snippet at a later point in time may be unaware of its outdatedness unless mentioned in a comment. Furthermore, it can be difficult to integrate the snippet without knowing the exact version of the library it is referencing. In this paper, we propose an automated approach to identifying ranges of Maven library versions that might have been used in a Java snippet on Stack Overflow. We use a prototype implementation of the approach to assess the overall outdatedness of Stack Overflow snippets with respect to the latest version of each referenced library available from Maven. We found a considerable number of snippets that use outdated library versions, which suggests that developers should be careful when adopting solutions from Stack Overflow.**

***Index Terms*—Stack Overflow, software libraries, method calls, Maven**

## I. Introduction

The Stack Overflow question and answer platform has become part of many a developer's toolbox [1]. It is used to share knowledge about technical issues in the form of code snippets that solve concrete problems. Referencing standard and third-party libraries, code snippets are seldom self-contained. Integrating code snippets in a software project therefore brings about the problem of identifying all referenced libraries, several versions of which might have evolved. Developers are therefore in need of an automated approach to identifying the exact versions of the referenced libraries that will complete the snippet.

In this paper, we propose an automated approach to identifying the versions of Maven libraries referenced in Java code snippets on Stack Overflow. The approach analyses the method calls in a snippet to compute a range of compatible library versions in which the called methods exist.

Tools that implement our approach not only facilitate integrating code snippets in software projects, but also provide insights about how outdated such a code snippet is. Indeed, older code snippets might reference API members that have been declared deprecated by newer versions of the library. While users can point out outdated solutions in comments, a recent study of comment-induced Stack Overflow updates by Soni et al. [2] found that snippet authors largely ignore them. As a result, many outdated snippets are never updated and thereby contribute to the spreading of misinformation, buggy code, or suboptimal solutions in general.

We therefore apply a prototype implementation of our approach on the snippets in the SOTorrent [3] dataset, and answer the following research questions:

$RQ_1$ **Which library versions are used in SO code snippets?** For each identified library in a snippet, we compute the range of compatible library versions and report on the range's proportion to all versions that have been released.

$RQ_2$ **How outdated are the libraries referenced in SO code snippets?** We consider the boundaries of each library version range, and study how outdated the snippet would be when it is completed with these versions.

## II. Related Work

Empirical software engineering research on Stack Overflow has grown along with the popularity of the question and answer platform. For example, Abric et al. [4] studied the fate of duplicate questions on Stack Overflow. Although duplicate questions are found to stem more likely from junior developers, they also receive answers that are different from those to the original question and hence provide additional insights. Zhang et al. [5] studied what users discuss in comments, and analysed the commenting dynamics on questions and answers. One of their findings is that the majority of comments are made after an answer has already been accepted. Soni et al. [2] explored how comments affect answer updates on Stack Overflow. They found that a large number of answers on Stack Overflow are never updated in spite of comments that warrant an update. Nikolaidis et al. [6] identified pieces of code in software projects that match the code provided in a Stack Overflow answer, and studied their effect on each project's technical debt. Ragkhitwetsagul et al. [7] detected code clones between the Qualitas project corpus [8] and Stack Overflow code snippets. They found 153 clones that have been copied from the Qualitas corpus into a Stack Overflow code snippet, 100 (66%) of which were outdated, and 10 of which were buggy and harmful for reuse. They also conducted a survey with 201 authors of Stack Overflow answers in which they found that 131 participants (65%) have ever been notified of outdated code and 26 (20%) rarely or never fix the code.

## III. Approach and Dataset

We use the SOTorrent dataset [3] as our source of Stack Overflow code snippets, the libraries.io dataset [9] as our source of library version information, and the Maven central repository[1] as our source of JAR files for library versions. These data sources are depicted in Figure 1, along with the necessary data processing and cleaning which we detail below.

### A. Data extraction and cleaning

Data extraction and cleaning proceeded as follows:

**1. Extracting code snippets:** Using the SOTorrent dataset of January $24^{th}$ 2020[2], we selected the answers related to the Java programming language by retaining those answers with a parent question tagged with $< java >$. Our approach relies on *import* statements to identify which library is used in a code snippet. We therefore also filtered out code snippets without any import statement and kept the others to be analysed in subsequent steps.

**2. Identifying libraries:** Using the libraries.io dataset [9] of January $12^{th}$ 2020, we extracted all library names with their versions and the date of release. For each library, we downloaded the JAR file of its latest release and extracted the fully qualified names of their classes (i.e., package path followed by the simple name).

**3. Selecting code snippets:** Based on the data from **step 2**, we identified all Stack Overflow code snippets that have at least one import statement referencing the pre-identified library classes. From each Stack Overflow post we selected only one code snippet, i.e., the snippet of the answer with the most votes. We found that there are $6,419$ libraries.io-indexed libraries that have at least one library import on Stack Overflow. These library imports are distributed across $19,444$ unique answers.

**4. Extracting library methods:** To identify the versions of a library that might have been used in a code snippet, we downloaded the JAR files of all versions of all libraries that we identified in **step 3**. This resulted in $50,574$ JAR files from the Maven central repository. For each JAR file (and hence library version), we use the *japicmp* [3] tool to extract all library method names along with the names of their hosting classes and their number of parameters. From all library versions, we extracted $324,944$ classes and $9,420,408$ unique methods. We found that only $5,545$ ($1.7\%$) of the extracted classes are explicitly imported into one or more Stack Overflow code snippets.

Table I summarises the resulting dataset of method calls and import statements extracted from code snippets.

**5. Extracting method calls:** Our approach uses the method calls in a snippet to identify the compatible version ranges of a library. While exploring the dataset obtained in **step 4**, we noticed that there are many import statements that can refer to more than one library. In fact, considering the whole dataset we surprisingly found, a considerable proportion of libraries that shares the same fully qualified class names (i.e., 20.2% of all classes can be found multiple times in 74.3% of all libraries). In many of those cases, the libraries appear to be forks or copies of other libraries. For example, the classe name `com.google.common.primitives.Floats` can be found in 46 libraries. One of these libraries is `org.hudsonci.lib.guava:guava` [4], clearly stating in its description that it is *a fork of Guava 14.0.1 for Hudson.* Filtering these ambiguous methods out led us to remove 75.8% of the method calls extracted from the code snippets. Furthermore, we found that there are many libraries having their package paths starting with the prefix "`java`" or "`javax`", both of which are used by the Java standard library. In order to enable identifying the exact libraries that are used within the snippets, we therefore filtered out all methods belonging to classes imported through ambiguous import statements. Next, we filtered out all method calls for which we could not find a matching method (using their name and number of parameters) among the methods of the JAR files extracted in the previous step. From $86,834$ method calls initially extracted, the filtering only retained $7,577$. These method calls belong to $435$ unique Maven libraries and are distributed across $1,760$ Stack Overflow answers. These libraries are used $1,901$ times (i.e., a library might be used in multiple snippets and a snippet might have multiple libraries). Table II depicts descriptive results of the final dataset of method calls used to identify library versions in Stack Overflow code snippets.

TABLE I
Number of unique libraries, import statements, method calls, and snippets identified and extracted from Stack Overflow.

| Unique libraries | Imports | Method calls | Snippets |
|---|---|---|---|
| 6,419 | 46,924 | 86,834 | 19,444 |

TABLE II
Number of unique libraries, Library usages, imports, method calls, and Stack Overflow snippets remaining after step 5.

| Unique libraries | Library usages | Imports | Methods | Snippets |
|---|---|---|---|---|
| 435 | 1,901 | 4,313 | 7,577 | 1,760 |

### B. Identifying library versions

Armed with the data extracted above, it is straightforward to identify the library versions that are compatible with a method call within a snippet that contains an import statement for its hosting class. As a snippet may have several import statements and method calls, we identify these ranges for each method call separately and take the intersection of the ranges in which all library methods called from the code snippet exist. This further improves the accuracy for a given Stack Overflow snippet and library.

## IV. Empirical Results

We now present the results for each research question by means of visualisations and statistics. To enable reproducibility of our work, we provide a publicly available replication package including the dataset and all code [5].

[1] https://mvnrepository.com
[2] https://zenodo.org/record/3627636
[3] https://siom79.github.io/japicmp/
[4] https://mvnrepository.com/artifact/org.hudsonci.lib.guava/guava
[5] https://doi.org/10.5281/zenodo.4568743

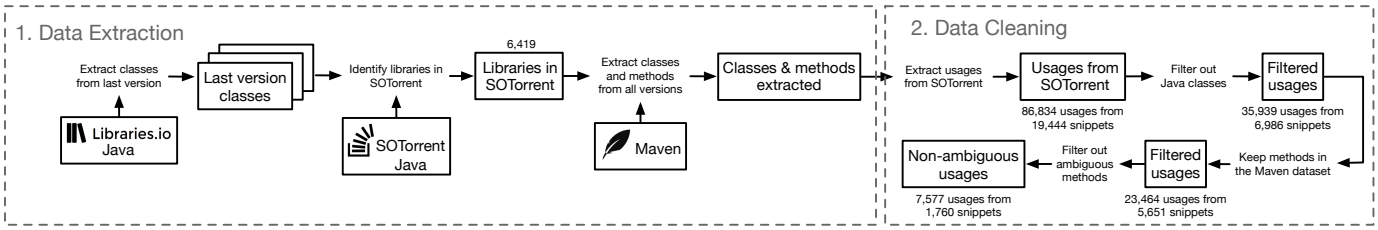


Fig. 1. Process of data extraction and filtering from different sources of information.

*$RQ_1$ Which library versions are used in SO code snippets?*

The goal of this research question is to identify the range of library versions that a method call in a code snippet is compatible with. For each method call extracted from Stack Overflow snippets, we identified the library to which it belongs, and then we identified the versions where such a method is part of the public interface of the library. From a starting number of $7,577$ method calls, we only found $1,507$ ($19.9\%$ of all methods) that have not existed in every released version of their libraries. The remaining library methods ($79.1\%$) exist in all versions, and thus we cannot say anything meaningful about the used versions in these cases as the used methods could be taken from any version in the library history.

The method calls in our dataset belong to $1,901$ not unique libraries. Excluding the calls to methods that exist in every library version, we found that the remainder belongs only to $447$ libraries which represents $23.5\%$ of the initial sample. These libraries are used in $417$ code snippets. For these calls, we could identify an actual range of compatible library versions. Figure 2 depicts a histogram of the number of libraries for which we could identify version ranges for calls in our Stack Overflow code snippets. The X-axis denotes the proportion of identified versions by these ranges to all versions of the library. For example, if a library $L$ has 10 versions and we identified 2 versions as compatible with the code snippet, then the proportion of versions in this case is $20\%$. For more than $23\%$ of the libraries, we could identify a subset of (possibly used) versions that represents only $10\%$ (i.e., red line in the figure) of all library versions. These libraries are used in 103 code snippets. This gives an idea of the relative size of the compatible version ranges.

**Finding:** Using our approach we could identify library version ranges for 19.9% of the method calls extracted from Stack Overflow. We found 76.5% of the studied libraries have methods that exist in all versions.

*$RQ_2$ How outdated are the libraries referenced in SO code snippets?*

Through this research question, we want to assess how much of a problem outdated library usage poses on Stack Overflow. We only focus on the $447$ libraries identified in $RQ_1$, i.e., libraries that are used in code snippets with method calls not present in every version of their library.

Every identified range of possible versions has two end points, the first version and the last version in the range.

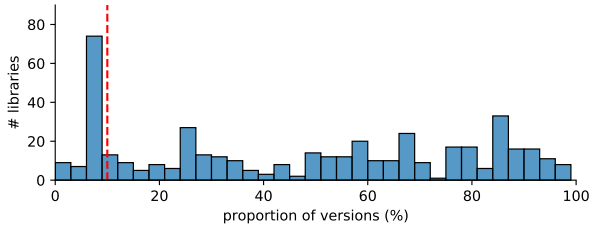


Fig. 2. Histogram representing the number of libraries and their corresponding proportion of identified versions.

We consider both end points to assess the *outdatedness* of each snippet. For each endpoint, we assume that it is the actual version used in the Stack Overflow snippet and then compare it to the latest available library version on Maven. We compute the difference between the latest version and the version assumed being in use, in terms of missed versions and number of days between their respective release dates. We assume that we do not know the date of the answer creation, and we study the outdatedness of the snippets as if they are going to be integrated today (i.e., at the analysis date).

**Best case scenario:** For $447$ libraries that are used in $417$ snippets, we found that $91$ of the libraries are outdated even if we assume that the library version used in the snippet is the last in the identified range of versions. Zooming in on the outdated libraries, we found that in the best case the median number of missed versions of these libraries is 12. Also, the median number of days between the used and the latest versions is $886$ days, i.e., nearly two and a half years.

**Worst case scenario:** Assuming that the used library version is the first in the identified version range, we found that $361$ (i.e., $80.7\%$) of the libraries are outdated. Zooming in on the outdated libraries, we found that in the worst case, the median number of missed versions of these libraries is $26$. Moreover, the median number of days between the latest version and the version assumed to be used by the snippet is $1,364$ days, i.e., more than 4 years.

The difference between the best and the worst case scenarios can also be seen in Figure 3, which shows a distribution in means of boxen plots of the number of missed versions and the difference in days between the latest version available on Maven and the version assumed to be in use.

**Finding:** Without considering the date when the answers were created, we found that their code snippets make use of outdated libraries, missing many recent releases.

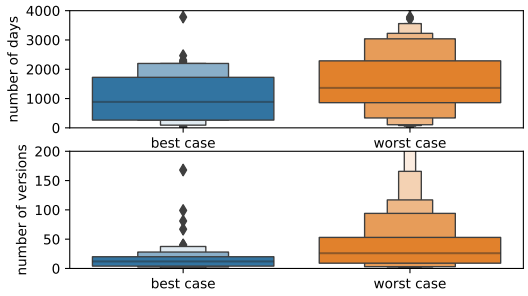


Fig. 3. Number of missed versions and days between the used library version and its latest available version, grouped by case scenario at the analysis date.

**Considering creation date.** We have also investigated the outdatedness of code snippets at the time when their answers were last modified:

- At the best case scenario, we found 115 outdated libraries. However, the level of outdatedness is lower than what we found when we studied snippets without considering their last modification date. The median number of missed versions of these libraries is 10, while the median number of days between the used and the latest versions is 518 days.
- At the worst case scenario, we found 329 outdated libraries. The median number of missed versions of these libraries is 17, while the median number of days between the used and the latest versions is 976 days.

**Finding:** Even at the time of their last update, Stack Overflow answers come with code snippets making use of outdated libraries.

## V. Discussion

To properly carry this approach we had to filter a considerable number of answers, libraries and method calls, which left us with a small number of data extracted from Stack Overflow. The filtering was mainly because there are many libraries that have the same paths to the same class names and it is not possible to identify a library version unless we first identify the exact library that is used in the code snippet. A manual inspection of many cases showed that forking is one of the main reasons behind this copy of source code. We believe that knowing the exact library name can help us to have more accurate results. It could be helpful to make use of the tags associated with Stack Overflow answers, since there are many developers that specify the used library in the tags. This will remain as part of our future agenda.

The high number of libraries presenting the same class paths is surprising. It could be interesting to study the problem of library copying and forking in the future. For example, to know to which extent the forks and exact copies of other libraries are used as part of open source projects.

However, using our approach we could identify ranges of possible library versions for method calls in Stack Overflow code snippets. In $RQ_1$, we identified the possible library versions of $19.9\%$ of the extracted method calls, leading us to reveal these versions for $23.5\%$ of all considered libraries. The rest of the called methods were found in all library versions.

In $RQ_2$ we verified whether the used libraries are outdated. We focused only on the code snippets that have method calls belonging to some specific library versions. We found some libraries that even in their best case scenario, are still outdated missing many recent releases. We also found code snippets having outdated libraries at their last modification date. This shows that our current approach can lead to useful insights about the *outdatedness* of libraries in Stack Overflow code snippets mitigating the risk of having break changes, bugs and security vulnerabilities. These findings also indicate that developers should be aware that there is a high chance that the code snippet they are copying from Stack Overflow is making use of outdated libraries.

## VI. Threats to validity

Our study only considered Java code snippets in Stack Overflow answers and Java libraries hosted on Maven. It is possible that some of the used libraries are not hosted in Maven. Thus, it might be possible that we missed some library usages. Furthermore, our results cannot be generalised to other libraries outside Maven or code written in other programming languages. Our approach, however, can be transposed to other library repositories and programming languages.

To enumerate candidates for the libraries that might have been used in Stack Overflow code snippets, our prototype relies on import statements and on the fully qualified names of library classes as they exist in the latest available version on Maven. It is possible that we missed some class names that were available in older versions but do not exist anymore in the latest versions.

To match the used methods in code snippets with methods in library versions on Maven, we relied on the method name, its host class, and the number of parameters it takes. We do not consider the types of parameters because we cannot identify all parameters' types from the code snippets. This can affect our results since there could be multiple matches for a call in case the corresponding method is overloaded with the same number of parameters, but with different types.

## VII. Conclusion

Stack Overflow is a widely known question and answer platform for developers, but it is not without its limitations. One of its limitations is the lack of tool-supported means for finding and flagging deprecated and outdated solutions. In this paper, we proposed an automated approach for identifying versions of Maven libraries used in Java code snippets. Using this approach, we could assess the *outdatedness* of such library usage and found a considerable number of Stack Overflow code snippets relying on outdated library versions. Our results suggest that developers should be wary of outdated library usage when adopting examples from Stack Overflow answers in their software projects.

## Acknowledgements

This research was partially funded by the Excellence of Science project 30446992 SECO-Assist financed by FWO-Vlaanderen and F.R.S.-FNRS.

## REFERENCES


[1] B. Vasilescu, V. Filkov, and A. Serebrenik, "StackOverflow and GitHub: Associations between Software Development and Crowdsourced Knowledge," in *2013 International Conference on Social Computing*. IEEE, 2013, pp. 188–195.

[2] A. Soni and S. Nadi, "Analyzing Comment-Induced Updates on Stack Overflow," in *2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR)*. IEEE, 2019, pp. 220–224.

[3] 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*, 2018, pp. 319–330.

[4] D. Abric, O. E. Clark, M. Caminiti, K. Gallaba, and S. McIntosh, "Can Duplicate Questions on Stack Overflow Benefit the Software Development Community?" in *2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR)*. IEEE, 2019, pp. 230–234.

[5] H. Zhang, S. Wang, T.-H. Chen, and A. E. Hassan, "Reading Answers on Stack Overflow: Not Enough!" *IEEE Transactions on Software Engineering*, 2019.

[6] N. Nikolaidis, G. Digkas, A. Ampatzoglou, and A. Chatzigeorgiou, "Reusing Code from StackOverflow: The Effect on Technical Debt," in *45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA'19)*. IEEE, 2019.

[7] C. Ragkhitwetsagul, J. Krinke, M. Paixao, G. Bianco, and R. Oliveto, "Toxic Code Snippets on Stack Overflow," *IEEE Transactions on Software Engineering*, 2019.

[8] E. Tempero, C. Anslow, J. Dietrich, T. Han, J. Li, M. Lumpe, H. Melton, and J. Noble, "The Qualitas Corpus: A Curated Collection of Java Code for Empirical Studies," in *2010 Asia Pacific Software Engineering Conference*. IEEE, 2010, pp. 336–345.

[9] J. Katz, "Libraries.io Open Source Repository and Dependency Metadata," Jan. 2020.