Summary

**JTeC: A Large Collection of Java Test Classes for Test Code Analysis and Processing**

**Arthur :** **Federico Corò , Roberto Verdecchia , Emilio Cruciani , Breno Miranda , Antonia Bertolino**

**Conference : In 17th International Conference on Mining Software Repositories (MSR ’20), October 5–6, 2020, Seoul, Republic of Korea**

**INTRODUCTION** :Test computerization has been effectively sought after since the 90's as an answer for lessen the significant expenses of programming testing and improve item quality . Zion Market Research assesses that the product test mechanization market will develop from the 16 Billion Dollars of 2016 up to 55. All the above examination endeavors request accessibility of a huge dataset of test code, to which the proposed techniques and devices can be applied. Anyway there doesn't yet exist such an assortment for prepared use by the network. As we ourselves are going to need such a dataset for approving our versatile way to deal with test prioritization , we might want to make accessible for the use of the network the test classes dataset that we gathered. At the hour of accommodation, the dataset, called JTeC , gives 2.5M+ test classes gathered from a lot of 31K+ tasks in Github, which starting now and into the foreseeable future the network taking a shot at test code investigation and handling can reuse without burning through further energy.

The JTeC dataset, and the quality sifting content which is given as supplement to the dataset, are accessible online in the JTeC group. In the rest of depict the technique followed to gather the dataset , its structure , and how it very well may be utilized for research . We finish up the paper indicating conceived future JTeC improvements.

**METHODOLOGY :**The first step of our process consists of indexing the public GitHub repositories, and is carried out in order to execute efficiently the subsequent phases of our process. In this step we first retrieve the name of the public repositories and the username of their creators. This is achieved via a query to the GitHub API3 specifying the unique identifiers of the repositories in an incremental fashion . As stopping criterion for the repository indexing process we adopt the number of repositories required to collect 2.5M test classes.

This number is approximated via a preliminary exploratory analysis, and amounts to approximately 31K repositories. Subsequently, once we obtain the list of repository names mapped to the usernames of their creators, we can launch a second query to the GitHub API in order to retrieve the programming languages associated with each repository. Once obtained a local copy of the indexed repositories mapped to their programming languages, we can effortlessly retrieve the URLs of the repositories developed in a specific language. Csv file created in the previous step to isolate the repositories developed exclusively in Java by inspecting the language tag of the indexed repositories, and subsequently combine their name with the username of their creators in order to identify the unique URL of the repositories.

**The final output of this step is the list of URLs corresponding to al**l Once we isolated the repositories developed in Java, we can identify the test classes of the repositories to store them locally. This is achieved by leveraging the standard naming conventions for Java test classes in Junit, the most popular Java unit testing framework . Specifically, we do this by selecting the source code files of the selected repositories whose name ends with «Test. Java» and «Tests

**Result :** They have presented the JTeC dataset that makes ready available to the community of software testing researchers a large collection of Java test classes useful for several potential purposes related to test code analysis and processing. Given the large scale, obtaining such information is clearly highly effort-intensive, and we consider that coarse criteria such as function coverage would be enough. Having this type of information could support benchmarking other regression testing techniques beyond black-box similarity-based ones. Additionally, to broaden the capabilities of JTeC, we envision to include in future versions also fault information of the identified test classes.

Providing this latter type of information, would enable researchers to have at disposal a large-scale real-world dataset containing the most important information related to test suites, enabling them to conduct a vast range of studies related to the research field of software testing.