An Analysis on "A Large Scale Study of Programming Languages and Code Quality in Github"

Critique of a study written by Baishakhi Ray, Daryl Posnett, Vladimir Filkov, Premkumar Devanbu

Arpit Garg
Faculty of ECMS
The University of Adelaide
Adelaide, SA, Australia
arpit.garg@.adelaide.edu.au

ABSTRACT

The following is a critique of the paper "A Large Scale Study of Programming Languages and Code Quality in Github" (Ray, et al., 2014). 'Authors' referred to the authors of this paper.

1 Is the research well-motivated?

Previously, comparison of programming languages was divided into three categories i.e. Controlled Experiment, Surveys, Repository Mining. In Controlled Experiment, developers were used to programming in different programming languages and outcomes were compared. In Surveys, based on different parameters surveys are conducted between developers to find the popularity of the languages and in Repository Mining, languages are compared to find out the reliability. However, these studies found no significant differences between the languages. Therefore, categorization of the languages will help the programmers to select the suitable category of the language according to the project requirements. As this creates a relationship between language, domain and defect type, and gives the dependency of languages based on bug categories. This will help the programming community to select the right tool for the job. It very well may be said that examination is all around well-motivated.

2 Is the paper well-framed in terms of related work?

In 'Related Work' section author divided the previous work into three sections Controlled experiment, Surveys and Repository Mining. In each section, the author explained the previous study and their shortcomings. For the respective section, the authors gave their solutions based on previous works. Overall, this paper was not fully dependent on previous studies. However, the authors introduce new categorization methods to make the comparison more generalized. A section like a methodology and a result are very well connected with related work, explaining each part from data collection to previous and current results and the methodology included getting the results. So, the paper is well framed, describing all the workings in each section in terms of related work.

3 Does the paper tell you what data was collected?

In the second section of the paper, the authors describe the Github projects and the languages collected and used. The author divides the 'Methodology' section into various parts and focused more on Data Collection. The main aim of the section 2.2 data collection was to select the Github Archive and to identify the top languages with the most popular projects and project history. In the section top, 19 programming languages are considered and some of the languages are removed as they are not considered to be general-purpose languages. The author separates the sections into six parts and provides a detailed explanation of each part. Furthermore, eighteen different evets of Github were considered and based on this table with summary is built depending upon project details, Total Commits and BugFix commits. As described above, the author fully described what data is collected, from where it is collected and why only this much data is collected, and all the others are ignored. The authors already described all the sources and information required, so nothing was. missing. Each section is well defining in the terms of data

4 Does the paper tell you how this data was analysed/interpreted and how the research question was answered

Based on data collected, the data is analyzed in a very efficient way to compare the different programming languages. Like in section 2.3, language is categorized in different classes and categories. Then after this in section 2.4, the projects considered are taken into 30 distinct different domains and the probability of each project based on these domains are calculated. It is very difficult to identify the common functionalities so these domains are assigned manually, and the naming of the domain was done accordingly. Section 2.5 was very important analysis as it helps the authors in categorizing bugs. Bugs are categorized based on Impact and cause and this classification is done in two steps. The first step involves automatically categorizes the messages based on keyword search. This step helps the authors to reduce false positives. The second step used the previously generated data as the training data to be supervised learning algorithm and the remaining bug fixes messages were treated as the test data and the Natural Language Processing is used for a bag of words which contains bug fix messages. In the last section of methodology, statistical methods were used. Negative Binomial Regression model was used to show the relationship between predictors and response and handle over-dispersion. Authors log-transform the dependent count variable to stabilizes the variance with the use of AIC and Vuong's test for some models. The conservative value of 5 was used to compute the variance inflation factor. Since the resulted factors were unbalanced, authors used weighted effects coding and Chi-Square test of independence was used to calculate the relationship between two-factor variables. This paper explains step by step how the data was analyzed with the methods and steps involved in interpretation.

5 Does the method appear suitable?

For describing the method suitability four aspects will be explained: relevancy, bias, validity and soundness.

5.1 Relevance

The method can answer the research question. In this research, the question aims to build a strong comparison between programming languages taking reference of the previous study. There are many languages so there must be some base of comparison between languages. Therefore, an effective way was used by authors to consider the previous studies and based on these comes up with a different categorization-based method. All the methods used for dividing the languages into different domains with the correlation between them were explained with methods used and every section is compared with previous study results. This is the main purpose of authors to make the results more accurate. Overall, the approaches and methods they used to answer the research question were relevant.

5.2 Bias

The paper should be unbiased, and authors should focus on fair results and conclusion. The authors of this paper try to remain fair while coming up with the results. All the domains of the languages were manually assigned and reviewed by the researchers. To detect the defect proneness also authors used Negative Binomial Regression model and tries to include every parameter possible like several developers and the size of the projects as these factors will directly influence the defect fixes. Rather considering individual language the classes were considered. While comparing languages authors found that the languages are identical in all aspects rather than coefficients and z-scores. By comparing all these, authors concluded that there is no strong association between languages, so they aggregate defects to make the results neutral. In section 2.6 all the outliers were controlled and filter them out as high leverage points. At the last part of the results, authors discuss all the relationship between language and bug category and the conclusion generalizes for all categories. They concluded that defect types are strongly associated with languages. They have also discussed the weakness that there is no relationship between domain and language defect. Authors try to make the paper as neutral as possible by considering all the major factors, including the weakness of the methods considered.

5.3 Validity

The author concluded paper with the result but these results should be valid. All the data used was taken from the Github archive but there are few threats to the validity. First, the bug database was not checked only keywords were considered. Assumptions were made based on language property while interpreting the language classes. Also, language properties associated with defect fixing commits. The scope of the data is general and the precision of detection was also moderate, the data used was new at the time of writing as compared to other sources and the methods are highly relevant with low bias. All these threats to validity were already expressed by authors which makes the method appropriate to answer questions. Although they were unable to quantify the effects of language type on usage, these all points were already defended by authors. Overall, it shows moderate results in terms of validity.

5.4 Soundness

It means the argument quality, related research based on logics and way of interpreted data. The whole data was taken from Github archive and for all the parameters and categorisation statistical approach were used which are discussed by authors in every part accordingly. The data was interpreted by finding the correlation between most of the data and most of the useless data was discarded. Every part of the paper was divided into answers based on different questions comprising of all the methods and algorithms used, specifying all the assumptions made. So the paper was very well presented in terms of soundness considering the quality of arguments.

6 Does the paper make sense? Is it well-presented and easy to read?

The whole paper was very well structured following the ACM format. Apart from this whole paper was logically related between each section like in section 2 was methodology which comprises of data collection, categorizing languages, identifying project domain categorizing bugs and Statistical methods. Based on this section, the next part of the paper results was explained followed by the comparison between the previous studies and the threats to validity. The authors brilliantly explain about all the statistical terms used. They try to describe this by giving the user information about the terms with the algorithms and model used. In every section, tables are used to demonstrate the data and comparison was done and heat maps were also used to describe the correlation. This information concludes that pape is very structured and easy to read.

References

Ray, B., Posnett, D., Filkov, V. & Devanbu, P., 2014. *A Large Scale Study of Programming Languages and Code Quality in Github.* Hong Kong, s.n.