A Simple NLP-based Approach to Support Onboarding and Retention in Open Source Communities

Christoph Stanik[1] Lloyd Montgomery[2] Daniel Martens [3] Davide Fucci[4] Walid Maalej[5]

Abstract: This work has been presented at the 34th International Conference on Software Maintenance and Evolution (ICSME) in Madrid, Spain [**?**][6].

Successful open source communities are constantly looking for new members and helping them become active developers. A common approach for developer onboarding in open source projects is to let newcomers focus on relevant yet easy-to-solve issues to familiarize themselves with the code and the community. The goal of this research is twofold. First, we aim at automatically identifying issues that newcomers can resolve by analyzing the history of resolved issues by simply using the title and description of issues. Second, we aim at automatically identifying issues, that can be resolved by newcomers who later become active developers. We mined the issue trackers of three large open source projects and extracted natural language features from the title and description of resolved issues. In a series of experiments, we optimized and compared the accuracy of four supervised classifiers to address our research goals. Random Forest, achieved up to 91% precision (F1-score 72%) towards the first goal while for the second goal, Decision Tree achieved a precision of 92% (F1-score 91%). A qualitative evaluation gave insights on what information in the issue description is helpful for newcomers. Our approach can be used to automatically identify, label, and recommend issues for newcomers in open source software projects based only on the text of the issues.

Keywords: open source software; onboarding; task selection; newcomers; machine learning; natural language processing

1 Research Design

When developers decide to start contributing to an Open Source (OSS) project they become newcomers to that project. In large OSS projects, newcomers face thousands of open and unresolved issues they have to choose from. The goal of this work is to identify issues in OSS projects that newcomers can resolve–tackling the barrier of finding a way to start.

[1] UHH, Informatik, Vogt-Kölln-Str. 30, 22527 Hamburg, Germany stanik@informatik.uni-hamburg.de
[2] UHH, Informatik, Vogt-Kölln-Str. 30, 22527 Hamburg, Germany montgomery@informatik.uni-hamburg.de
[3] UHH, Informatik, Vogt-Kölln-Str. 30, 22527 Hamburg, Germany martens@informatik.uni-hamburg.de
[4] UHH, Informatik, Vogt-Kölln-Str. 30, 22527 Hamburg, Germany fucci@informatik.uni-hamburg.de
[5] UHH, Informatik, Vogt-Kölln-Str. 30, 22527 Hamburg, Germany maalej@informatik.uni-hamburg.de
[6] Open access link: `https://arxiv.org/abs/1806.02592`

doi:10.18420/se2019-19

Dataset. Our dataset contains issues of three large OSS projects, namely Qt, Eclipse, and LibreOffice. In total, we collected 225,000 issues from 2001 to 2017. For our research, we consider only the title and the description of the issues.

Machine Learning Experiments. In a series of machine learning experiments[7] we used natural language processing (NLP) to classify issues into "can be resolved by a newcomer" or "cannot be resolved by a newcomer". As for NLP text features, we used tf-idf, the sentiment of the text, as well as the number of words in the issues. In the experiments, we compared the performance of four classifiers including hyper-parameter tuning using grid search.

Qualitative Investigation. To better understand how mentors–experienced OSS contributor that help newcomers finding solvable issues–and newcomers themselves select issues to work on, we conducted eight semi-structured interviews and performed thematic analysis on a random sample of issues.

2 Results

Our results show that Random Forest can achieve, on average, a precision of 79% in identifying issues newcomers can resolve. We optimized the classification benchmarks towards precision as we think that it is more important to be sure that a newcomer can resolve an issue than covering all issues–which would introduce a lot of noise (high recall), and therefore includes more issues newcomers cannot resolve.

The qualitative analysis provides us with insights on how we might be able to improve our approach further. The interviews revealed that issues that contain, e.g., *code snippets* or *steps to reproduce* are more helpful for newcomers as this information hints towards where and how to start working in the code-base. The thematic analysis resulted in similar findings but further shows that experienced contributors not only work on more complex issues but also on issues newcomers can resolve.

[7] replication package: `https://mast.informatik.uni-hamburg.de/replication-packages/`