Classification of Issues

Bharadwaj Sharma Kasturi
Golisano College of Computing and Information Sciences
Rochester Institute of Technology
bk5953@rit.edu

Niranjana Sathish Avilery
Golisano College of Computing and Information Sciences
Rochester Institute of Technology
na6322@rit.edu

Omkar Sanjay Narkar
Golisano College of Computing and Information Sciences
Rochester Institute of Technology
on9164@rit.edu

I. Introduction

Software maintenance and evolution is vital for the success of a software project. It deals with the task to eradicate the potential defects in the code and in addition to this performs the task of understanding the users' transpiring needs and evolving the code based on it [1]. Tools such as issue tracking systems help support these tasks by providing facilities to efficiently signal, manage, and address tickets or potential problems arising in software systems. However, in several projects, especially the popular ones, tens or hundreds of issues are reported daily. This leads to heavier workloads for developers in turn leading to an increase in the software maintenance costs. For the projects hosted on GitHub an issue is reported by merely providing a title and an optional description. Since these issues can be of different types, GitHub offers a customizable labeling system to tag these issue reports. Manually tagging each issue does help in the issue processing but it is a tedious and time-consuming process nonetheless [2], which is why labels are barely used on GitHub [3] [4].

II. Existing Studies

Studies in the past have validated the use of traditional text mining methods to automatically detect and classify bug reports [5] but it only took the description part of the reports into account. Since the contents of the description may be unstructured texts in natural languages, Zhou et al. proposed that structural information can help in building a finer prediction tool [6], [7]. The study also discussed an approach that combines text mining and data mining methods to classify the bug reports. The paper proposes three stages starting with using the bug report summary to categorize into different levels of the corrective report done by multinomial Naïve Bayes Classifier followed by using the structured features of bug reports together with the features extracted from the Naïve Bayes Classifier. It gathers the initial text classifier's output, locates the matching bug report in the repository, and merges it with additional characteristics from the same report. This was able to avoid the noise caused by misclassification and offer better performance for bug prediction. Kallis et al. proposed a study that used Ticket Tagger to predict the issues in Github [8] and compared it with the J48 algorithm to classify texts. Ticket Tagger [9] is a GitHub application that uses Machine Learning to classify and label the types of reports submitted to each issue. Ticket Tagger uses a pre-trained fastText model to categorize based on three main categories: bug report, enhancement, and question. 10-fold cross-validation was applied to both Ticket Tagger and the J48 algorithm and it was found that Ticket Tagger was able to obtain higher performance metrics for each label compared to the J48 algorithm.

III. Problem Statement

In this project, we aim at identifying the type of issue that has occurred by analyzing the report submitted by the developer. The title and the description of the issue reported are utilized to classify the type of issue as a bug report, feature request or a question [10]. As reacting to these issues manually is a lot of effort and cost of maintaining the software is also to be considered, the classification of issues is automated using Machine Learning algorithms utilizing the Title and description as the input from the users.

IV. Limitation of existing studies

Ticket Tagger, is a tool developed to automatically identify the issue types by processing the reports title and body [11]. The current approach lacks the preprocessing of text data we use for classification algorithms, if that is done stop words and irregular patterns in texts could be prevented and make the algorithm more efficient. Parameter tuning can be done for a better classification of issues [12]. Stop words like how or what tends to assign the label as a question irrespective of the nature of the actual issue class. The language could be made more consistent which would enhance the ticket tagger performance.

V. Proposed Solution

From the limitations of previously existing solutions it can be observed that using raw data from the Github without any preprocessing has been a limitation. So, applying the machine learning algorithms on the data which has been preprocessed, i.e. the stop words like 'why', 'how' or 'what' can be eliminated and language Can be made more consistent which would enhance the performance of the model and give even better scores.

REFERENCES

[1] A. Di Sorbo, G. Grano, C. Aaron Visaggio, and S. Panichella, "Investigating the criticality of user-reported issues through their relations with app rating," *Journal of Software: Evolution and Process*, vol. 33, no. 3, p. e2316, 2021.

[2] Q. Fan, Y. Yu, G. Yin, T. Wang, and H. Wang, "Where is the road for issue reports classification based on text mining?" in *2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. IEEE, 2017, pp. 121–130.

[3] J. Cabot, J. L. C. Izquierdo, V. Cosentino, and B. Rolandi, "Exploring the use of labels to categorize issues in open-source software projects," in *2015 IEEE 22nd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*. IEEE, 2015, pp. 550–554.

[4] T. F. Bissyandé, D. Lo, L. Jiang, L. Réveillere, J. Klein, and Y. Le Traon, "Got issues? who cares about it? a large scale investigation of issue trackers from github," in *2013 IEEE 24th international symposium on software reliability engineering (ISSRE)*. IEEE, 2013, pp. 188–197.

[5] G. Antoniol, K. Ayari, M. Di Penta, F. Khomh, and Y.-G. Guéhéneuc, "Is it a bug or an enhancement? a text-based approach to classify change requests," in *Proceedings of the 2008 conference of the center for advanced studies on collaborative research: meeting of minds*, 2008, pp. 304–318.

[6] N. Bettenburg, R. Premraj, T. Zimmermann, and S. Kim, "Extracting structural information from bug reports," in *Proceedings of the 2008 international working conference on Mining software repositories*, 2008, pp. 27–30.

[7] Y. Tian, D. Lo, and C. Sun, "Drone: Predicting priority of reported bugs by multi-factor analysis," in *2013 IEEE International Conference on Software Maintenance*. IEEE, 2013, pp. 200–209.

[8] R. Kallis, A. Di Sorbo, G. Canfora, and S. Panichella, "Predicting issue types on github," *Science of Computer Programming*, vol. 205, p. 102598, 2021.

[9] R. Kallis., A. Di Sorbo, G. Canfora, and S. Panichella, "Ticket tagger: Machine learning driven issue classification," in *2019 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, 2019, pp. 406–409.

[10] P. Floris and H. V. Harald, "How to save on software maintenance costs, omnext white paper," *SOURCE 2 VALUE*, 2010.

[11] Z. Liao, D. He, Z. Chen, X. Fan, Y. Zhang, and S. Liu, "Exploring the characteristics of issue-related behaviors in github using visualization techniques," *IEEE Access*, vol. 6, pp. 24 003–24 015, 2018.

[12] S. Herbold, A. Trautsch, and F. Trautsch, "On the feasibility of automated prediction of bug and non-bug issues," *Empirical Software Engineering*, vol. 25, no. 6, pp. 5333–5369, 2020.