Classification for Ticket Tagger: Machine Learning Driven Issue with Practical Swarm Optimization algorithm

Afrah Alharbi, Azzah Alamri, Fatimah Albrahimi Aljohani, Salma Alyami

College of Computer and Information Science
Imam University Riyadh, Kingdom of Saudi Arabia
afsaharbi@sm.imamu.edu.sa, azhomari@sm.imamu.edu.sa, falbrahimi@sm.imamu.edu.sa, amasalma@sm.imamu.edu.sa

Abstract: Software maintenance is crucial for software projects evolution and success: code should be kept up-to-date and error-free, this with little effort and continuous updates for the end-users.

Keywords: Issue reports management, Labeling issues, Text classification , PSO

I. INTRODUCTION

Software maintenance involves tasks for mitigating potential defects in the code, as well as for evolving it according to the users' emerging needs [1]. Thus, it is crucial for the success of software projects. Issue tracking systems are tools to support these tasks by providing facilities to efficiently signal, manage, and address tickets or potential problems arising in software systems. The tracking of tasks and issues is a common part of modern software engineering, e.g., through dedicated systems like Jira and Bugzilla, or integrated into other systems like GitHub Issues[2]. software developers are required to timely react to issues reported in issue trackers and solve such issues by investing the lowest possible effort, to keep the costs related to software maintenance low [3].

In projects hosted on GitHub, issue submitters report new issues by simply providing a title and an optional description of the issue. As issues of different types (e.g., asking questions, proposing features, signaling bugs) and quality could be submitted, GitHub also offers a customizable labeling system that can be used by developers to tag issue reports (e.g., by specifying the issue category or the related development tasks). Such labeling has positive effects on issues processing [4], making it easier for their management and prioritization [5]. labels assigned to issues help to classify and filter the reports.

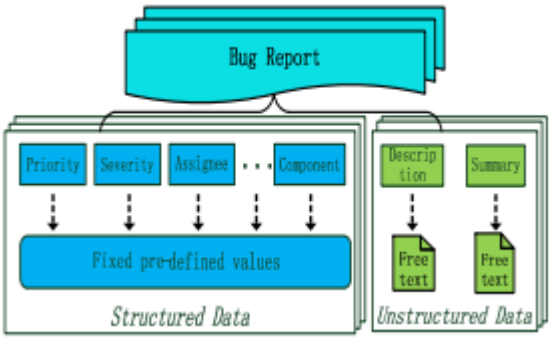

Figure 1: Bug report fields illustration

To help maintainers dealing with issue processing, we developed Ticket Tagger [6], a tool able to automatically label issue reports. Differently from previous approaches aimed at automatically identifying issue types [7, 8], since GitHub (according to its lightweight structure) does not provide any structured information about such issues, our tool exclusively relies on the textual features contained in the titles and descriptions of the reports to enable the automated labeling of them, immediately after they are submitted. This is beneficial for developers interested to handle new issues [5]. Ticket Tagger automatically predicts the labels to assign to issues, with the aim of stimulating the use of labeling mechanisms in software projects, this to facilitate the issue management and prioritization processes.

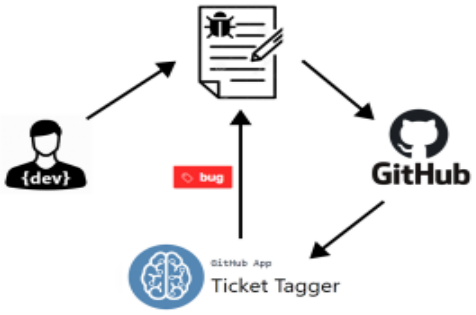

Figure 2: Ticket Tagger issue labeling process

In our paper ticket tagger classify issues by text. Text classification is a process that applies a set of algorithms to convert the unstructured information to organized text and analyze the organized data objects [9]. Feature selection is one of the most important factors which can influence the classification accuracy rate. Feature selection is to select a minimal sized relevant features from the original large set of features for the purpose of building robust, accurate and fast models [10].

To better address the problems we use Particle swarm optimization (PSO) as feature selection. efficient global search technique is needed. Particle swarm optimization (PSO) [11] is a relatively recent Evolutionary Computation (EC) technique based on swarm intelligence and are well-known for its potential global search ability. Among versatile population-based swarm intelligence and evolutionary algorithms [12] that aim at solving global optimization of numerical problems, Particle Swarm Optimization (PSO) [13,14] is one of the most widely applied and esteemed. PSO has some particles (solutions) which combine self-experiences with social experiences. The PSO algorithm has a main nested loop that terminates when the total number of function evaluations (iterations) exceeds a certain limit or a minimum error threshold is achieved[15].

Our contribution Within this article, we want to investigate if machine learning models for the prediction of issue types can be improved by practical Swarm optimization algorithm to deal with the problem happen and make a ticket tagger on it.

The remainder of the paper is organized as follows: Section II introduces previous studies. Section III presents methodology. The experiments result with performance evaluation and discussion are explained in Section IV. Section V presents our conclusions

II. RELATED WORKS

Previous studies presented several approaches to automatically categorize issues posted in bug tracking systems.

A. TICKET TAGGER

Kallis et al.[1] they propose Ticket Tagger, a GitHub app analyzing the issue title and description through machine learning techniques to automatically recognize the types of reports submitted on GitHub and assign labels to each issue accordingly. they empirically evaluated the tool's prediction performance on about 30,000 GitHub issues. their results show that the Ticket Tagger can identify the correct labels to assign to GitHub issues with reasonably high effectiveness.

Antoniol et al. [7] they investigates whether the text of the issues posted in bug tracking systems is enough to classify them into corrective maintenance and other kinds of activities. Results from empirical studies performed on issues for Mozilla, Eclipse, and JBoss indicate that issues can be classified with between 77% and 82% of correct decisions.

Herzig et al. [16] introduced six different issue categories – bug, feature request, improvement request, documentation request, refactoring request, and others – and demonstrated that often developers and maintainers assign the wrong issue category to the reports.

Zhou et al. [8] they propose a hybrid approach by combining both text mining and data mining techniques of bug report data to automate the prediction process. Also combined structured data with unstructured free-text data to train a classifier able to predict with high accuracy if a bug report is actually a bug or another kind of issue. Comparative experiments with previous studies on the same data—three large-scale open source projects— consistently achieve a reasonable enhancement (from 77.4% to 81.7%, 73.9% to 80.2% and 87.4% to 93.7%, respectively) over their best results in terms of overall performance.

B. PARTICAL SWARM OPTIMIZATION ALGORITHM

Abualigah et al. [17] proposed a novel feature selection method, namely, feature selection method using the particle swarm optimization (PSO) algorithm (FSPSOTC) to solve the feature selection problem by creating a new subset of informative text features. The results revealed that the proposed method (FSPSOTC) enhanced the effectiveness of the text clustering technique by dealing with a new subset of informative features.

Bai et al. [9] Presents an approach using a novel two stage method for text feature selection, where with the features selected by four different filter ranking methods at the first stage, more irrelevant features are removed by PSO to compose the final feature subset. The experimental results show that the proposed two-stage method can substantially reduce the dimensionality of the feature space and improve the classification accuracy.

III. METHODOLOGY

IV. RESULTS AND DISCUSSIONS

V. CONCLUSION

In this work, we presented swarm optimization algorithm works with Ticket Tagger to make a classification to the bugs, and automatically assigns suitable labels to issues opened on GitHub projects. Indeed, fellow developers who desire to improve the issue maintenance process through the automated classification enabled by the tool, can easily integrate Ticket Tagger into their repositories.

REFERENCES

[1] Kallis, R., Di Sorbo, A., Canfora, G., & Panichella, S. (2021). Predicting issue types on GitHub. *Science of Computer Programming*, *205*, 102598.

[2] Herbold, S., Trautsch, A., & Trautsch, F. (2020). On the feasibility of automated prediction of bug and non-bug issues. *Empirical Software Engineering*, *25*(6), 5333-5369.

[3] P. Floris, H. Vogt Harald, How to save on software maintenance costs, omnext white paper, SOURCE 2 VALUE.

[4] Z. Liao, D. He, Z. Chen, X. Fan, Y. Zhang, S. Liu, Exploring the characteristics of issue-related behaviors in github using visualization techniques, IEEE Access 6 (2018) 24003–24015.

[5] J. L. C. Izquierdo, V. Cosentino, B. Rolandi, A. Bergel, J. Cabot, Gila: Github label analyzer, in: International Conference on Software Analysis, Evolution, and Reengineering, SANER, 2015, pp. 479–483.

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

[7] G. Antoniol, K. Ayari, M. Di Penta, F. Khomh, and Y. Guéhéneuc, "Is it a bug or an enhancement?: a text-based approach to classify change requests," in Proceedings of the 2008 conference of the Centre for Advanced Studies on Collaborative Research, October 27-30, 2008, Richmond Hill, Ontario, Canada, 2008, p. 23.

[8] Y. Zhou, Y. Tong, R. Gu, and H. C. Gall, "Combining text mining and data mining for bug report classification," Journal of Software: Evolution and Process, vol. 28, no. 3, pp. 150–176, 2016. [Online]. Available: https://doi.org/10.1002/smr.1770

[9] Bai, X., Gao, X., & Xue, B. (2018, July). Particle swarm optimization based two-stage feature selection in text mining. In *2018 IEEE Congress on Evolutionary Computation (CEC)* (pp. 1-8). IEEE.

[10] J. Tang, S. Alelyani, and H. Liu, "Feature selection for classification: A review," *Data Classification: Algorithms and Applications*, p. 37, 2014. [Online]. Available: http://eprints.kku.edu.sa/id/eprint/170

[11] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *IEEE International Conference on Evolutionary Computation (CEC)*, 1998, pp. 69–73.

[12] J. Del Ser, E. Osaba, D. Molina, X.S. Yang, S. Salcedo-Sanz, D. Camacho, S. Das,P.N. Suganthan, C.A. Coello Coello, F. Herrera, Bio-inspired computation: where westand and what's next, Swarm Evol. Comput. 48 (2019) 220–250.

[13] R.C. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in:Proceedings of the 6thInternational Symposium on Micromachine Human Science,Nagoya, Japan, IEEE, Piscataway, NJ, USA, 1995, pp. 39–43.

[14] J. Kennedy, R.C. Eberhart, Particle swarm optimization, Perth, Australia, in:Proceedings of the IEEE International Conference on Neural Networks, IEEE,Piscataway, NJ, USA, 1995. IV 1942-1948.

[15] Sedighizadeh, D., Masehian, E., Sedighizadeh, M., & Akbaripour, H. (2021). GEPSO: A new generalized particle swarm optimization algorithm. *Mathematics and Computers in Simulation*, *179*, 194-212.

[16] misclassification impacts bug prediction," in International Conference on Software Engineering, ICSE '13, 2013, 2013, pp. 392–401. [Online]. Available: https://doi.org/10.1109/ICSE.2013.6606585

[17] Abualigah, L. M., Khader, A. T., & Hanandeh, E. S. (2018). A new feature selection method to improve the document clustering using particle swarm optimization algorithm. *Journal of Computational Science*, *25*, 456-466.