CASTR: Assisting Bug Report Assignment Recommender Creation

Disha Devaiya, John Anvik, Farjana Yeasmin Omee, Meher Bheree

Department of Mathematics and Computer Science
University of Lethbridge, Alberta, CANADA
E-mail: `devaiya86@gmail.com, [john.anvik, omee, bheree]@uleth.ca`

Issue tracking systems are used to make a software development process more manageable, especially for a geographically dispersed team. However, for each bug report, a decision-making process called *bug report triage* needs to be performed. A common bug report triage decision is the assigning of a developer to a specific bug report.

Bug report triage can take significant time and resources. Bug report assignment recommenders have been proposed for reducing the workload of a project member. However, creating a recommender is complex, as project members have to perform several steps such as data preparation and selection of a machine learning algorithm. Although previous work has sought to find specific answers for aspects of the assignment recommender creation process, to the best of our knowledge, only a few other works have examined assisting with this recommender creation process.

CASTR (Creation Assistant for Supporting Triage Recommenders) [1, 2] is a platform-independent multi-tier web application. It allows a project member to analyze the dataset using a graphical representation and also assists in configuring project-specific parameters for a machine learning algorithm. This demonstration shows how to use CASTR to create a bug report assignment recommender, which consists of the following steps:

1. *Download the dataset from the project's issue tracking repository.* CASTR provides an interface that allows a user to set parameters, such as a date range and number of reports, for downloading a dataset using the Bugzilla REST endpoints. For the demonstration, we will be using a data set from a large open-source project, such as Firefox, PlasmaShell or LibreOffice.

2. *Configure and create the recommender.* Information about the collected dataset is displayed by the `Configuration` tab. This tab assists a user in performing data filtration by setting project-specific heuristics for report labelling and a minimum activity threshold value for which labels (i.e. developers) to recommend. Also, the user selects a machine learning algorithm (SVM, Multinomial Naïve Bayes, C4.5, or rule-based) to use for recommender creation. Finally, the user can select an algorithm to handle data imbalance (oversampling using SMOTE, undersampling using clusters, or manual oversampling). Once the user has set the recommender configuration parameters, the user clicks the "Create Recommender" button and is taken to the *Analysis* tab.

3. *Analysis of recommender.* The `Analysis tab` presents evaluation results for the created recommender. It displays the Top-1, Top-3, and Top-5 precision and recall values for a testing set of 100 randomly selected reports that are not used for training. Also, the `Analysis` tab displays an example of possible labels and predictions for some randomly selected bug reports from the testing set. A `History` panel allows a user to select a previous recommender configuration. The `Confusion Matrix` tab displays for each developer name, the distribution of correct and incorrectly predicted class values from the testing set.

4. *Retrain the recommender.* If needed, the user can then return to the `Configuration` tab, adjust the configuration parameters, and create a new recommender. This process continues until the user is either satisfied with the created recommender, or the user has determined that an assignment recommender cannot be created with a high enough accuracy to benefit the project.

This work was funded by the Natural Sciences and Engineering Research Council of Canada.

DOI reference number: 10.18293/SEKE2021-207

References

[1] D. T. Devaiya. Castr: A web-based tool for creating bug report assignment recommenders. Master's thesis, University of Lethbridge, Lethbridge, Alberta, CANADA, 2019.

[2] D. T. Devaiya, J. Anvik, and M. Bheree. Evaluating a tool for creating bug report assignment recommenders. In *Proceedings of the 33rd International Conference on Software Engineering and Knowledge Engineering*, 2021.