Issue-Labeler: an ALBERT-based Jira Plugin for Issue Classification

Waleed Alhindi
Department of Information Technology
Prince Mohammad Bin Fahd University
Khobar, Saudi Arabia
waleed_rh_hindi@hotmail.com

Abdulrahman Aleid
Department of Information Technology
Prince Mohammad Bin Fahd University
Khobar, Saudi Arabia
aleid.2009@gmail.com

Ilyes Jenhani
Department of Software Engineering
Prince Mohammad Bin Fahd University
Khobar, Saudi Arabia
ijenhani@pmu.edu.sa

Mohamed Wiem Mkaouer
Department of Software Engineering
Rochester Institue of Technology
Rochester, United States
mwmvse@rit.edu

***Abstract*— Issue labels are key drivers in software maintenance as they dictate the prioritization, organization, and ultimately the resolution of encountered issues. Consequently, mislabeling issues result in inefficient prioritization, which compromises the resolution process of these issues. Thus, to increase the accuracy and effectiveness of issue labeling in software maintenance, this paper proposes "Issue-Labeler": an automated issue labeler plugin for Jira[1], which utilizes a deep neural language model to predict an issue's type based on its title and description. Specifically, the plugin would classify an issue into three types: BUG, IMPROVEMENT, and NEW FEATURE. The issue-labeler plugin was implemented by fine-tuning Google's pre-trained ALBERT language model, using 35,889 labeled issue reports extracted from 77 projects. The plugin showed an average F1-score of 0.75, 0.58, and 0.67, respectively, for the BUG, IMPROVEMENT, and NEW FEATURE issues. The plugin will provide developers with a tool that recommends issue labels to, in turn, optimize the process of tagging and resolving these issues. Video of tool setup and runtime is available: https://youtu.be/mi2FwaXNrR4. Tool Webpage: https://issue-labeler.github.io/issue-labeler-site/. Replication package: https://github.com/issue-labeler/.**

Keywords— Issue Classification, Jira, Software Maintenance, Plugin, ALBERT

I. INTRODUCTION

Issue tracking is a fundamental activity of software development, where developers create reports, in the form of textual descriptions, of what they anticipate as actions to perform in the project. These entries can be used to report a task, a bug, a description of a feature request, or even a question about anything related to the project. Due to the importance of tracking issues, they have been adopted by development communities. For instance, Jira is a widely adopted project management software used for tasks and workflow management, requirements and test-case management, and, pertinently to this paper, issue tracking [1]. As of 2020, Jira had been adopted by over 65,000 companies and 180,000 customers across 190 countries with 83% of Fortune 500 companies cited as utilizing some form of Jira product [2]. Since trackers receive various types of issues, developers need to label them based on their content (e.g., BUG, FEATURE), as part of their management and resolution. This labeling task can quickly become a bottleneck, especially for projects experiencing high volumes of issues, where triaging upcoming issues can be subjective, error-prone, and time-consuming [3].

As such, this project aims to provide developers with an issue-labeler Jira plugin that automatically predicts/verifies the label of a given issue. This plugin is augmented with a model that indicates the issue's type based on its title and description. The plugin may also help developers preview what is brought by the title and/or description they have used and can quickly update their 'unclear' title and/or description to point directly toward a specific issue category. The plugin also helps project triagers with the association of appropriate code reviewers. Specifically, the plugin's prediction engine is an issue-classification deep-learning language model implemented by fine-tuning Google's pre-trained ALBERT language model. Introduced by Google in 2019, ALBERT, which stands for A Lite BERT, is a smaller version of the original BERT language model [4].

ALBERT achieves a smaller size due to the factorization of BERT's embedding parameters leading to an 80% reduction in the parameters of the projection block at the expense of only a minor drop in performance [5]. While it is known that BERT and other of its variants, such as seBERT and RoBERTa, provide promising results regarding issue classification [11,12,13,16], the large size of these models becomes a challenge for their adoption in practice. That said, ALBERT still retains BERT's fundamental architecture in implementing self-supervised learning of language representations. BERT, which stands for Bidirectional Encoder Representation from Transformers, is a deep-learning language model introduced by Google in 2018 [6].

BERT incorporates a transformer-based architecture to implement the attention mechanism [7]. Attention, in this context, refers to the breakdown of inputs into smaller segments and analyzing each segment with regard to every other segment [8]. Transformers apply attention by splitting a sentence into its composite words and then analyzing each word against each other to establish the context of the sentence as a whole. Up until recently, attention was primarily implemented by utilizing recurrent neural networks (RNNs) and long-short-term-memory networks (LSTMs) [9], both sequentially analyze inputs leading to a directional limitation in applying attention techniques [6]. On the other hand, transformers are non-directional in nature, facilitating the implementation of attention techniques bi-directionally and increasing the comprehensiveness of contextually analyzing inputs [9]. Due to these capabilities, transformers have seen recent widespread use in Natural Language Processing (NLP) tasks such as text summarization, translation, and classification [6]. As such, ALBERT, A Lite BERT, had been

[1] https://www.atlassian.com/software/jira

