Data-Driven Approach for Log Instruction Quality Assessment

Vrushabh Vilas Wadnere
vw1937@g.rit.edu
Rochester Institute of Technology
Rochester, New York, USA

Rohaan Nadeem
rn7823@g.rit.edu
Rochester Institute of Technology
Rochester , New York, USA

Robert Gentilucci
rag8467@g.rit.edu
Rochester Institute of Technology
Rochester New Yourk, USA

ACM Reference Format:
Vrushabh Vilas Wadnere, Rohaan Nadeem, and Robert Gentilucci. 2023. Data-Driven Approach for Log Instruction Quality Assessment. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 1 page. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In the ever-evolving landscape of IT, the roles of developers and system operators are distinct yet intricately intertwined. Developers craft lines of code, while operators execute these codes, often with limited visibility into their inner workings. The crucial bridge connecting these two worlds is a humble yet indispensable tool: log messages. These messages serve as a means of communication between developers and operators, helping pinpoint the causes of issues and facilitating the reporting of bugs during system operation. However, despite their vital role, there exists a noticeable absence of universally accepted guidelines for crafting high-quality log instructions. These instructions typically consist of structured text, including log levels (such as "info" or "error"), static text, and dynamic variables. This disparity in standards can lead to issues like assigning inappropriate log levels for minor events or providing insufficient information in static text. In this context, logging emerges as an integral practice in modern software development, producing software logs that play a pivotal role in debugging, maintenance, and system analysis. In this introductory exploration, we delve into the world of log instructions, their significance, and the pressing need for quality standards in their creation and utilization. The existing literature on quality properties and logging practices within the realm of software development and maintenance is expansive and diverse, offering valuable insights into critical aspects of software engineering. This body of research encompasses a range of studies, with a notable emphasis on the examination of quality properties and logging practices. Pioneering studies have meticulously quantified the prevalence and benefits of logging in various programming languages, such as C/C++, Java, and Android systems. These studies have not only shed light on developers' efforts to fine-tune log instructions but have also uncovered intriguing aspects, including the relative stability of log instruction locations and the dynamic nature of log instructions throughout a system's lifecycle. Additionally, comprehensive field studies have explored the economic aspects of logging, providing a comprehensive perspective from both developers' and researchers' viewpoints. The convergence of findings across different programming languages has paved the way for contemporary research endeavors, focusing on cross-examining software systems while assessing the quality of log instructions. Another significant research avenue revolves around automated logging enhancement methodologies, categorized into two primary groups: those addressing the where-to-log issue, i.e., log instruction placement, and those tackling the what-to-log challenge, involving the selection of pertinent logging information. The latter category further subdivides into log message generation, placement of relevant variables, and log-level suggestion. These innovations represent a burgeoning field of research aimed at streamlining and optimizing logging practices in software development, demonstrating the dynamic and evolving nature of this vital area within the software engineering domain. However, this research paper addresses the fundamental problem of achieving high-quality log instructions in software development. Despite the availability of quality guidelines, developers frequently encounter difficulties in crafting effective log statements, leading to incomplete or incomprehensible log messages. Furthermore, the absence of mechanisms within logging frameworks to monitor log instruction quality places the responsibility solely on developers, resulting in issues such as inaccurate log level assignment and inadequate linguistic structure, which negatively impact software quality and developer productivity. These challenges are particularly pronounced in complex software environments with diverse developer backgrounds and programming languages. As such, the paper recognizes the need for an automated approach to log quality assessment to navigate the complexities introduced by heterogeneous software systems and developer practices. In response to the previously mentioned constraint, this paper endeavors to present an automated method for evaluating the quality of log data originating from any software system. To achieve this we will try replicating the QuLog is an automated system-agnostic solution for evaluating log instruction quality. It uses deep learning to assess log-level correctness and linguistic structure sufficiency based on the static text of log instructions. QuLog comprises three components: log instruction preprocessing, a deep learning framework with separate networks for each quality property, and a prediction explainer. In the offline phase, it learns from diverse software systems, and in the online phase, it provides log level and linguistic quality predictions along with word importance scores as recommendations to help developers improve log instructions.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn