

Enhancing Flutter App Development: Addressing Configuration and Compatibility Bugs Using Large Language Models

Rajaguru R.M.S.N.^{1*}, Chathumini K.G.L.¹, Kumara B.T.G.S.², and Wijerathna P.M.A.K.³

¹*Department of Computing and Information Systems, Faculty of Computing, Sabaragamuwa University of Sri Lanka, rmsnrajaguru@std.appsc.sab.ac.lk*

¹*Department of Computing and Information Systems, Faculty of Computing, Sabaragamuwa University of Sri Lanka, lohara@foc.sab.ac.lk*

²*Department of Data Science, Faculty of Computing, Sabaragamuwa University of Sri Lanka, kumara@foc.sab.ac.lk*

³*Department of Software Engineering, Faculty of Computing, Sabaragamuwa University of Sri Lanka, ashansa@foc.sab.ac.lk*

In today's world, there are cross-platform frameworks are widely available, The frameworks like Flutter, allow the developer to develop an app from a single codebase. But there are some major issues like debugging and compatibility finding in Flutter framework. These issues lead to prolonged debugging and lesser app reliability. The proposed model will lead to overcome those mentioned challenges. The proposed model used advanced LLMs including GPT-4o, Claude Sonnet 3.5, and Gemini 2.0 Flash, combined with RAG capabilities. The data was collected from multiple sources like Stack Overflow, GitHub, and Flutter documentation. The dataset was cleaned and preprocessed by removing low-scoring answers, filtering incomplete questions, and applying text sanitization techniques to ensure structured and relevant inputs for analysis. A fixed set of prompt templates to 350 Stack Overflow questions across all the LLMs were applied to commence the evaluation process. The 25 most important questions were identified through verification via cosine similarity and precision correction between its responses. It was able to determine the two models' efficacy in finding configuration bugs and addressing those bugs. The current work involves integrating an RAG pipeline and using a vector database for better retrieval and response generation. Early evidence hints at the enhanced accuracy/precision of an RAG-enhanced LLM when that LLM is a standalone model for better Flutter configuration. In conclusion this proposed model presents a modular framework to integrate RAG-enhanced LLMs for bug detection and fixing. The framework requires much lesser debugging efforts along with more reliability of the app. Additionally, it can also adopt the fast-evolving Flutter framework. By filling a significant gap in cross-platform development literature, it helps advance AI-assisted debugging and improve the development workflows of Flutter apps.

Keywords: *Compatibility Issues, Configuration Bugs, Flutter, Large Language Models, Retrieval-Augmented Generation*