A Sinhala Based Programming Assistance Tool For Java Programmers

K. S. N. Athukorala
(Reg. No.: MS23060180)

A THESIS
SUBMITTED TO
SRI LANKA INSTITUTE OF INFORMATION TECHNOLOGY
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE IN INFORMATION TECHNOLOGY
(ENTERPRISE APPLICATIONS DEVELOPMENT)

December 2024

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Dr. Dilshan De Silva

Approved for MSc. Research Project:

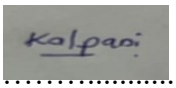
MSc. Programme Co-ordinator, SLIIT

Approved for MSc:

Head of Graduate Studies, FoC, SLIIT

DECLARATION

This is to certify that the work is entirely my own and not of any other person, unless explicitly acknowledged (including citation of published and unpublished sources). The work has not previously been submitted in any form to the Sri Lanka Institute of Information Technology or to any other institution for assessment for any other purpose.

Sign: 

K. S. N. Athukorala

Date: 14th December 2024

ABSTRACT

A Sinhala Based Programming Assistance Tool for Java Programmers

K. S. N. Athukorala
MSc. in Information Technology (EAD)
Supervisor: Dr Dilshan De Silva
December 2024

This research presents an extensive programming assistance tool, which is implemented especially for native Sinhala-speaking Java programmers. The tool provides several features to overcome challenges faced by Java programmers, including real-time code generation, diagram creation based on user queries and repository assistance for Java-related projects. It utilizes advanced NLP techniques and large language models (LLMs) to convert Sinhala queries into Java code, visualize code flow through diagrams, generate detailed reports from repository data and provide answers according to user given queries. Additionally, the tool incorporates a transformer-based translation model that converts Sinhala queries into English for code and diagram generation. The tool demonstrates high reliability and technical accuracy, with an overall translation accuracy of 95.79%, indicating strong alignment with reference translations. These capabilities significantly enhance the accessibility and productivity of Sinhala speaking developers. The system provides precise and contextually relevant responses to programming queries related to specific repositories by integrating large language models (LLMs) with a Retrieval-Augmented Generation (RAG) architecture. Core functionalities include repository assistance, enabling users to clone, query, and obtain targeted insights into codebases. Response generation is powered by LLMs, including GPT-3.5-turbo, GPT-4-turbo, GPT-4o-mini, GPT-4o, and ChatGPT-4o-latest. To evaluate the system's effectiveness, the tool was tested using the RAGAS (Retrieval-Augmented Generation Answer Scoring) framework, assessing performance across three key metrics: answer relevance, answer similarity, and answer correctness. This study demonstrates how such tools can democratize iii programming education, overcome language barriers, and increase inclusivity in technical learning, ultimately contributing to the growth of Sri Lanka's technology sector.

ACKNOWLEDGEMENT

I would like to thank and convey my appreciation to everyone who contributed to the success of this research. I am grateful to the Faculty of Graduate Studies and Research and all our lecturers for providing us with the greatest training and information needed to accomplish this research and get significant professional experience.

I am extremely grateful to Dr Dilshan De Silva for his consistent encouragement and mentorship. His advice, honest review, and encouragement have been invaluable throughout the research process. I genuinely thank him for the chances and freedom he provided, which have helped me grow both academically and personally.

TABLE OF CONTENTS

DECLARATION ii
ABSTRACT iii
ACKNOWLEDGEMENT iv
TABLE OF CONTENTS v
List of Figures viii
List of Tables ix
Chapter 1 Introduction 1
1.1 Research Problem 4
1.2 Objectives 5
1.2.1 Main Objective 6
1.2.2 Sub Objectives 6
1.3 Research Questions 11
1.3.1 How does a Sinhala-language programming assistance tool enhance the comprehension and coding efficiency of Sinhala-speaking Java developers? 11
1.3.2 How does the inclusion of diagram generation within the tool improve code structure comprehension and debugging accuracy for Sinhala-speaking Java developers? 12
1.3.3 To what extent does the repository assistance feature in the tool enhance comprehension and navigation of complex codebases for Sinhala-speaking developers? 12
1.3.4 How effectively does the tool narrow the gap between Sinhala-speaking and English-speaking developers in terms of code comprehension and productivity? 13
1.3.5 What specific usability challenges do Sinhala-speaking developers encounter with the tool's interface and features, and how can these be addressed to enhance user experience and accessibility? 14
Chapter 2 Background 16
2.1 Languages 16
2.1.1 Sinhala Language 16
2.1.2 English Language 16
2.2 Related Tools and Technologies 16
2.2.1 Java Programming Language 16
2.2.2 Natural Language Processing (NLP) 17
2.2.3 Large Language Models (LLMs) 17
2.2.4 Transformer Model 18
2.2.5 GitHub 19
2.2.6 ChatGPT 20
2.2.7 LangChain Framework 21
2.2.8 React Framework 21

Chapter 3 Literature Review 23
3.1 Programming Assistance Tools- Related Research 23
3.2 Translation-Related Research 24
3.3 Git Repository Management with LLMs-Related Research 26
3.4 Research Gap 32
Chapter 4 Methodology 38
4.1 Back-end API Development 39
4.1.1 Fast API Integration in Diagrams and Code Generation Handling 39
4.1.2 Fast API Integration in Repository Management 42
4.2 RAG Architecture 52
4.3 Transformer-Based Sinhala-English Translator 53
4.4 Front End Development 58
Chapter 5 Results 68
5.1 To develop a Transformer model to translate Sinhala to English and vice versa 68
5.2 To develop a system that implements a seamless diagram generation and code generation facility for Java programmers who are more comfortable with Sinhala 75
5.3 To enhance repository assistance and integration features within the system, enabling users to effectively query in Sinhala and comprehend code repositories hosted on popular platforms like GitHub 76
5.4 Evaluating Tool Usability, Effectiveness, and Performance 78
5.4.1 User Testing and Quality Assurance Integration 79
Chapter 6 Discussion 87
6.1 Relation to Research Objectives 87
6.1.1 To develop a Transformer model to translate Sinhala to English and vice versa 87
6.1.2 To develop a system that implements a seamless diagram generation and code generation facility for Java programmers who are more comfortable with Sinhala 87
6.1.3 To enhance repository assistance and integration features within the system, enabling users to effectively query in Sinhala and comprehend code repositories hosted on popular platforms like GitHub 88
6.1.4 Evaluate Tool Usability, Effectiveness, and Model Performance 88
6.2 Contribution to the Body of Knowledge 90
6.3 Novelty 91
6.4 Technical Soundness 91
6.5 Practical Applications and Potential Impact 92
6.6 Future Research and Development 92
Chapter 7 Conclusion 93
Chapter 8 Future Work 95
8.1 Enhance supporting other programming languages 95
8.2 Applying to Other Spoken Languages 95

8.3 Applying Localization for Enhanced Usability 95
References 97
Appendix 100
Appendix 1: User Interface of the welcome page in this system 100
Appendix 2: User Interface of the Repository Assistance in English 101
Appendix 3: User Interface of the initial view of the repository assistance 102
Appendix 4: Initial view of Epoch 1 in transformer model training. 102
Appendix 5: Model Accuracy 103

List of Figures

Figure 1 System diagram of the proposed Sinhala-based programming assistance tool 38
Figure 2 Python code for code generation API 40
Figure 3 Python code for mermaid code generation API 40
Figure 4 Python code for repository clone API 43
Figure 5 Python code for process of clone and indexing repository 43
Figure 6 Python code for process of clone repository 44
Figure 7 Python code for process of loading and indexing repository 45
Figure 8 Python code for asking questions about the repository 47
Figure 9 Python code for answering questions using repository data 48
Figure 10 Python code for process of preparing prompt template 48
Figure 11 Python code for Contextual Setup and Question Formatting 49
Figure 12 Python code for Searching and Answering the Question 50
Figure 13 Python code for Formatting and Document Retrieval 50
Figure 14 Python code for similarity check and identify most similar documents 52
Figure 15 Transformer model architecture 54
Figure 16 Illustration of encoders and decoders in a transformer model, demonstrating the process with a Java prompt to check if a number is prime, accompanied by explanatory text in Sinhala. 54
Figure 17 Illustration of the attention mechanism in a Transformer model, emphasizing its ability to capture long-range dependencies and selectively focus on various segments of an input sequence 55
Figure 18 Diagram of a Transformer model, showcasing the progression from input embeddings through multi-head attention, feed-forward networks, and positional encoding, culminating in output probabilities via a softmax function. 56
Figure 19 User interface displaying a Java code corresponding to a given question in Sinhala 59
Figure 20 React code generating Java code and diagrams based on a user's input 63
Figure 21 React component integrates Mermaid 63
Figure 22 React component is designed to create a chat interface for interacting with a repository 64
Figure 23 Distribution of words per sentence in English and Sinhala 70
Figure 24 Example for image of Generated diagram corresponding to a given question in Sinhala 75
Figure 25 Project Explanation in Repository Assistance System in Sinhala 77
Figure 26 Project Explanation in Repository Assistance System in English 77

List of Tables

Table 1 Comparative Analysis of Programming Assistance Tools & Translators for Sinhala and English... 34
Comparative Analysis of Programming Assistance Tools and Translators for Sinhala and English
Comparison of Research Tools and Methods for Git Repository Management and Large Language Model (LLM) Applications 36
Table 3 Breakdown of sentences across the training, validation and testing datasets 72
Table 4 Examples of Translated Sentences..... 73
Table 5 Comparison of LLM Model Performance 80
Table 6 Usability Testing Summary 86