AI-Driven Code Comment Quality Assessment and Its Impact on Software Complexity

J.C.N. Nagodavithana
(MS24010108)

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

December 2025

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.

Prof. 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is entirely my own and not of any other person, unless explicitly acknowledged (including citation of published and unpublished sources).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: J.C.N Nagodavithana
J. C. N. Nagodavithana

Date: 29/12/2025

ABSTRACT

J. C. N. Nagodavithana
MSc. in Information Technology
Supervisor: Prof. Dilshan I. De Silva
December 2025

Code comments play a critical role in software readability and maintainability. However, poorly written, redundant, or misleading comments can increase software complexity and hinder developer productivity. This research proposes a novel AI-driven framework for assessing code comment quality using a Comment Quality Index (CQI), which combines both structural and semantic features. Structural scoring was implemented through heuristic methods, while semantic scoring leveraged transformer-based models, specifically BERT, to capture the meaning and relevance of comments. To validate the approach, heuristic and AI-based scores were compared against developer-rated comments collected via a survey. Additionally, the study introduces CQI value bands to classify comments as poor, average, or good, providing actionable insights for developers. Statistical analyses, including ANOVA and correlation tests, confirm the effectiveness and reliability of the proposed scoring framework. For the AI component, the BERT model was fine-tuned on 90 developer-rated comments, showing consistent training loss reduction across epochs. The fine-tuned model was saved and applied for inference on unseen comments, demonstrating its ability to generalize and provide real-time quality assessment. The results indicate a strong alignment between AI-predicted scores and human evaluations, highlighting the potential of AI-assisted comment analysis to enhance software quality. Finally, the research explores the integration of this framework into an IDE plugin, enabling developers to receive immediate feedback on comment quality during code development. Overall, the study provides a comprehensive methodology for automated comment quality evaluation, combining empirical validation, AI-based semantic analysis, and practical implementation in modern software engineering environments.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my supervisor, Prof. Dilshan I. De Silva, for his invaluable guidance, continuous support, and constructive feedback throughout the course of this research. His expertise and encouragement were instrumental in shaping both the direction and quality of this study.

I would also like to extend my appreciation to the faculty and staff of the Faculty of Computing, SLIIT, for providing the necessary resources and a conducive environment for conducting this research.

Special thanks go to my colleagues and friends who offered their assistance, insightful discussions, and motivation during the various stages of this project.

Finally, I am deeply grateful to my family for their unwavering support, patience, and encouragement, which has been a constant source of strength throughout my MSc journey.

TABLE OF CONTENTS

DECLARATION ii
ABSTRACT iii
ACKNOWLEDGEMENT iv
TABLE OF CONTENTS v
List of Figures vii
List of Tables viii
Chapter 1: Introduction 1
1.1 Background 1
1.2 Research Problem & Motivation 2
1.3 Research Questions & Objectives 3
1.4 Scope of the Study 3
1.5 Significance of the Study 4
1.6 Thesis Organization 5
Chapter 2 Literature Review 6
2.1 AI Models for Comment Generation 6
2.2 Quality Assessment Tools and Frameworks 7
2.3 Comment-Code Consistency 8
2.4 Code Complexity and Static Analysis 9
2.5 Identified Research Gaps 21
2.6 Positioning of Current Study 21
Chapter 3 : Methodology 22
3.1 Research Design & Approach 22
3.2 Dataset Description 22
3.3 Data Preprocessing & Labeling 24
3.3.1 Derivation of the Comment Quality Index (CQI) 25
3.4 AI Model Setup & Training 26
3.5 Complexity Analysis Methods 28
3.6 Developer Survey & Empirical Validation 29
3.7 Ethical Considerations 30
Chapter 4 : Results & Analysis 32
4.1 Data Analysis Overview 32
4.2 Model Performance Metrics 32
4.3 Complexity Analysis Results 34
4.4 Comparison of AI-Predicted Scores vs Developer Ratings 36

4.4.1 Correlation Overview 36
4.4.2 Semantic Model vs Human Ratings 36
4.4.3 Heuristic CQI vs Human Ratings 37
4.4.4 Readability vs Human Ratings 37
4.4.5 Average Scores Comparison 38
4.4.6 Statistical Validation (ANOVA) 39
4.4.7 Practical Implications of Results 40
4.4.8 Reflections on Developer Survey Validation 40
4.4.9 Limitations in Results Interpretation 40
4.4.10 Bridging Results to Discussion 41
4.4.11 Examples of Model Misclassification 41
Chapter 5 : Discussion 42
5.1 Interpretation of Results 42
5.2 Implications for Software Quality & Maintainability 43
5.3 Limitations of the Study 45
5.4 Discussion on Semantic-Based Score vs CQI 46
Chapter 6 : Conclusion & Future Work 49
6.1 Summary of Key Findings 49
6.1.1 Contributions of the Study 50
6.2 Contributions to Research & Practice 51
6.3 Recommendations for Future Work 51
6.4 Final Research Outcome 53
References 55
Appendices 57
Appendix 1: Survey Questionnaire 58
Appendix 2: Developer Rating Survey (Comment Quality Evaluation) 65
Appendix 3: ANOVA Output 67
Appendix 4: Pearson Correlation Matrix 68
Appendix 5: Sample Code Snippets and Dataset Structure 70
Appendix 6: Extra Figures 72
Appendix 7: Author's Publications 74

List of Figures

Figure 1.Evaluation of the researching aspects in code comments..... 15
Figure 2. The citations for published research papers in the area of code comments..... 16
Figure 3. The Comparison of the Code Quality Assessment Techniques..... 18
Figure 4.Correlation Heatmap of CQI, Fog Index, BERT predictions, and Developer Ratings..... 36
Figure 5. Scatter plot of BERT Predictions versus Developer Ratings 37
Figure 6.Scatter plot of CQI values versus Developer Ratings 37
Figure 7. Scatter plot of Fog Index values versus Developer Ratings 38
Figure 8. Comparison of average scores for BERT, CQI, Fog Index, and Developer Ratings 38

List of Tables

Table 1.Comparison of initially reviewed 23 research papers 14
Table 2. The Challenges and the related solutions..... 17