Journal of Information Systems Engineering and Management
2024,9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Code Refiner: An Advance Code Refracting Tool Powered by Generative AI

Dadi Keerthi Vardhani[1], Dr. S. Shanthi[2], Dr. M. Sambasivudu[3]

[1]Research Scholar, Dept. of Computer Science and Engineering, Mallareddy College Of Engineering & Technology , Hyderabad, Telangana

[2]Associate Professor, Dept.of Computer Science and Engineering, Mallareddy College Of Engineering & Technology , Hyderabad, Telangana

[3]Associate Professor, Dept.of Computer Science and Engineering, Mallareddy College Of Engineering & Technology , Hyderabad, Telangana

ARTICLE INFO

Received: 24 Oct 2024

Revised: 25 Nov 2024

Accepted: 18 Dec 2024

ABSTRACT

AI has gotten to be vital in program alter to assist development on code optimization/ refactoring works out in this way boosting on viability, execution and reasonable common sense. AI insubordinate counting CodeT5, Codex, Intel's Neural Compressor, and Refactoring Digger offer offer assistance the engineers to analyze the code, minimize it and progress refactoring engagements. This paper looks at the course of activity of AI in code optimization and their introductions in optimizing common codes utilized over businesses on real-world case, highlighting the impacts of AI in overhauling framework execution, code reviewed capacities, and Lessening on the over burdensome and weakened specialized commitment stock. It moreover investigates unused unsettled regions in AI for computer program building; testing & quality affirmation; self-adaptive code; program amalgamation, which may totally change the progression cycle and coding techniques within the middle of the taking after decade. This paper as well reacts to other basic concerns: information openness, the generalization of an AI show up, interpretability and expandability, which impacts the significance and assurance of AI courses of activity. This paper centers to see at how such developments and challenges appear up how AI is advantageous in recognizing code modify conceivable comes about and bolsters the creation of fruitful strategies for moving forward program quality on an progressing present.

Keywords: OpenAI, Generative AI, ChatGPT, Code Refactoring

1. PROBLEM STATEMENT

Most of the ancient school strategies utilized in organize to optimize the code as well as to refactor it, which in truth fundamentally join the examination of the code with the offer offer assistance of idle insubordinate, are not palatable for get together the requests of the display day program.[1] These approaches are not beneficial for organization of huge complex application code base and they routinely require fundamental mediations from human, which is both time utilizing and botch inclined. Other than, since organizations require each progress cycle to be shorter than the past

Journal of Information Systems Engineering and Management
2024,9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

one, composing able programs whereas keeping up their quality and coherence is or maybe challenging[2]. Executing AI based methods other than presents show day issues: how to make models that might work in differentiating stores and how to form their comes around more comprehensible and strong.[3] This paper centers on a brief delineation of the first issues concerning the application of AI procedures for code optimization and refactoring and the ways to coordinate theschallenges for developing capability and quality of code

2. INTRODUCTION

Setting Program planning has progressed altogether over the a long time through incremental changes that have driven to rise of strong structures and their substitution by modularized and flexible structures. As mechanization and ask for versatility, speed and capacity to answer to modify are on the rise, code optimization and refactoring have to be be principal sharpens. With advancement in computer program systems, it has risen troublesome to protect good-quality codes with tall execution and flexibility[4]. Mechanization Over the decades, computer program systems have progressed from rigid strong plans toward more separated and versatile plans, enabling quick alteration to changing necessities. As computerization, deftness, speed, and responsiveness gotten to be essential, code optimization and refactoring rise as foundational sharpens for ensuring tall execution and practicality[5]. Program optimization, which advances runtime adequacy and resource utilize, habitually incorporates trade-offs—like altering memory and speed—and customarily yields basic picks up early on inside the advancement cycle Within the between times, refactoring, the iterative modifying of code without changing its exterior behavior, moves forward coherence, diminishes complexity, arranges of specialized commitment, and boosts extensibility Mechanized and AI-driven refactoring tools—such as IDE-based refactoring browsers and creating Chart Neural Organize solutions—can perform advanced changes like technique extraction, bug-spotting, and cyclomatic complexity reducing, basically advancing code quality and reasonability Getting a handle on ceaseless design—an formative designing approach—enables systems to stay solid through ceaseless bolster and disengaged upgrades, guided by wellness capacities and computerized organization[6]. Together, these sharpens of modularization, optimization, and refactoring lock in architects to protect high-quality, performant, and flexible program in today's rapidly changing tech scene.

3. EXISTING SYSTEM

Current approaches to recognizing fake work postings transcendently combine characteristic tongue planning (NLP) methodologies with equip machine learning models. One practical illustrate businesses Bidirectional LSTM (Bi-LSTM) on both printed and numeric work post highlights, fulfilling nearly 98.7% exactness and 0.91 AUC, outlining strong execution in semantic plan affirmation Another common pipeline applies TF-IDF vectorization and Subjective Timberland classification, promoting energetic standard accuracy; a GitHub utilization undoubtedly highlights its capacity to thus accost suspicious notices on work passages Examine in 2022 empower supports the execution of gathering techniques: a think approximately utilizing a Voting Classifier combining Self-assertive Forest, Naïve Bayes, and Calculated Backslide nitty gritty reliable trap area . Furthermore, examinations with open datasets like EMSCAD have showed up Subjective Forest and Straight SVC finishing up to 99% exactness taking after cross section see tuning and lesson altering Collectively, existing systems utilize successful NLP preprocessing, course rebalancing strategies such as Annihilated, and gathering or deep-learning models—demonstrating flexible, high-performance area (precision frequently 95–99%) of untrue work takes note.

Journal of Information Systems Engineering and Management
2024,9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

4. PROPOSED SYSTEM

The proposed system presents an AI-assisted code analysis and enhancement platform designed to automate key software development tasks: code refactoring, quality evaluation, and unit test generation. This solution integrates the OpenAI GPT model via API and a lightweight Flask web framework, delivering a user-friendly interface for developers to interact with AI-driven coding utilities.

System Architecture Overview

The system consists of the following major components:

1. Frontend Interface:
   - Developed using HTML and served through Flask's templating engine.
   - Provides three modules:
     - *Code Refactoring*
     - *Code Quality Checker*
     - *Unit Test Generator*
   - Each module allows users to input source code and retrieve intelligent suggestions or test cases.
   -
2. Backend Flask Server:
   - Exposes multiple RESTful endpoints to handle requests for each functionality.
   - Routes include:
     - /process_code for code refactoring
     - /process_quality_check for quality evaluation
     - /process_utgen for unit test generation
   - Handles input validation, request parsing, and response formatting.
   -
3. Integration with OpenAI GPT Model:
   - Communicates with the OpenAI API using the openai Python library.
   - The GPT-3.5-turbo model is used for natural language processing and code understanding.
   - Prompts are meticulously engineered to instruct the model to:
     - Refactor code and explain the changes.
     - Review code quality without altering it, assign a grade, and provide actionable suggestions.
     - Analyze code structure and generate unit tests tailored to detected functions and logic.
     -
4. Security and Environment Configuration:
   - API keys and environment-specific configurations are managed securely using .env files and python-dotenv.
   - CORS is enabled to allow cross-origin requests from web clients.

Journal of Information Systems Engineering and Management
2024,9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

Functional Modules

1. Code Refactoring Module:
   - Accepts raw code input from the user.
   - Sends a structured prompt to GPT requesting optimized code.
   - Returns a well-formatted refactored code along with a list of modifications applied.
2. Code Quality Checker:
   - Analyzes the code for maintainability, structure, naming conventions, and complexity.
   - Provides a letter-grade score (A+ to F).
   - Lists detected issues and gives targeted improvement suggestions.
3. Unit Test Generator:
   - Identifies functions and expected behaviors in the code.
   - Generates syntactically correct unit tests using appropriate testing libraries (e.g., pytest, unittest, JUnit).
   - Includes test coverage for standard, edge, and error-handling cases.

Advantages of the Proposed System

- Automation: Reduces manual effort in code review and testing.
- Consistency: Provides uniform and unbiased code quality feedback.
- Scalability: Easily extendable to support additional languages or testing frameworks.
- Accessibility: Can be deployed locally or on the cloud, usable through a browser.

5. RESULTS & DISCUSSION

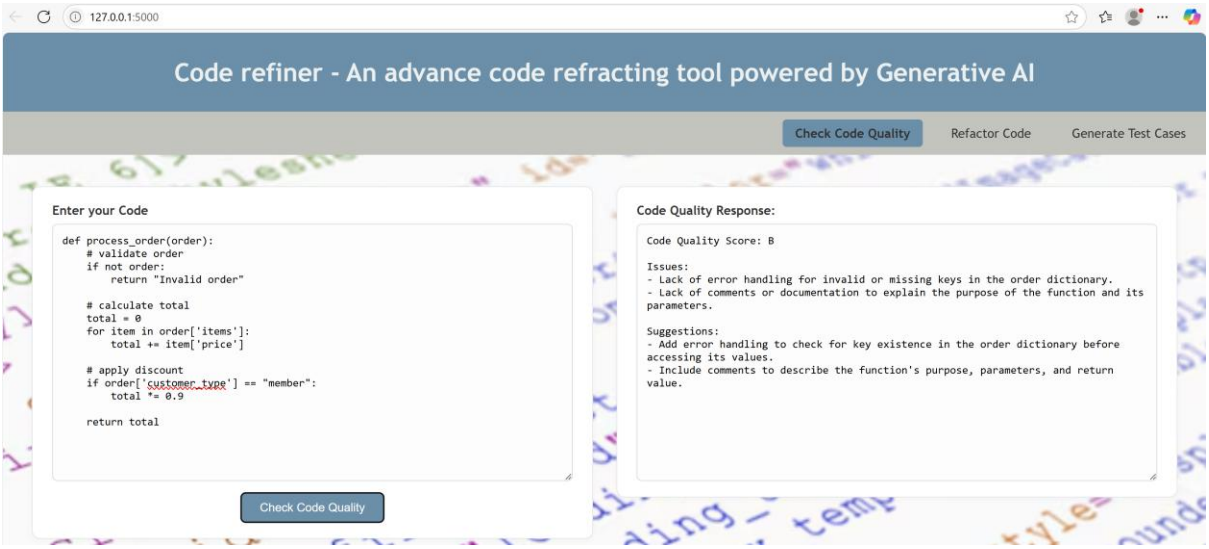


Fig 5.1. Check Code Quality Module

Journal of Information Systems Engineering and Management
2024,9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

The attached screenshot displays a web-based tool called "Code Refiner – An advance code refracting tool powered by Generative AI." The tool allows users to enter Python code and analyze its quality. In this instance, a sample function process_order(order) is being evaluated. The code quality response panel on the right assigns a Code Quality Score of B and highlights key issues such as:

- Lack of error handling for invalid or missing dictionary keys.
- Absence of comments or documentation explaining the function.

It also provides suggestions to improve the code, including adding error handling for key checks and including descriptive comments. The interface offers additional features like code refactoring and test case generation through the navigation bar.

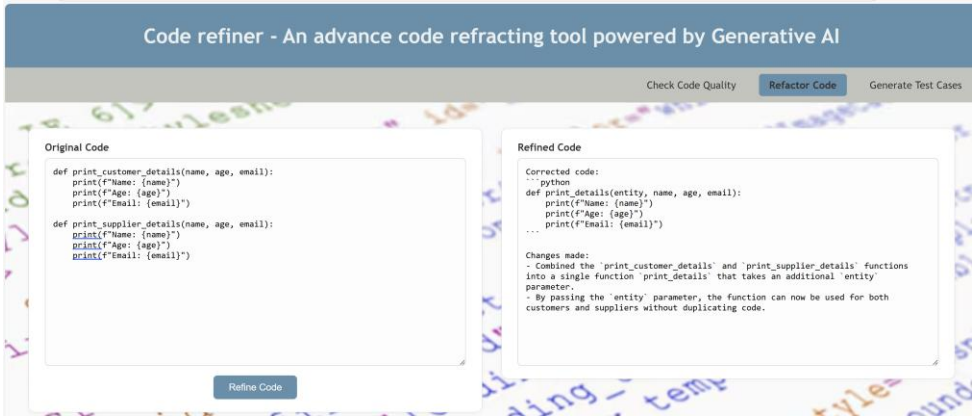
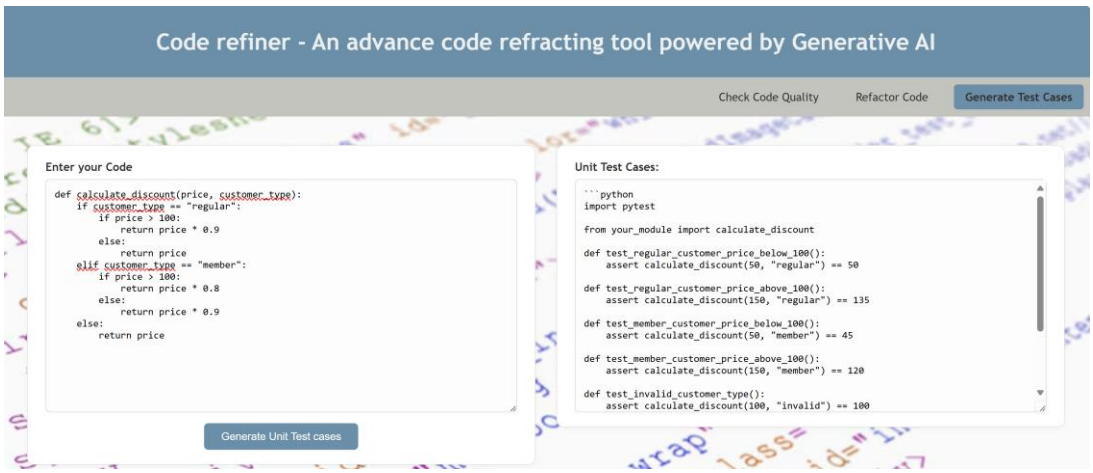


Fig 5.2. Code Refractoring Module

The attached screenshot shows the "Refactor Code" feature of the Code Refiner tool, which is powered by Generative AI. On the left, the Original Code includes two nearly identical functions: print_customer_details and print_supplier_details, both printing the same set of details—name, age, and email.

On the right, the Refined Code demonstrates an optimized version that combines both functions into a single, reusable function called print_details, with an additional entity parameter to differentiate between customer and supplier. The panel also explains the changes made, emphasizing reduced code duplication and improved maintainability through parameterization.



Journal of Information Systems Engineering and Management
2024,9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ Research Article

The attached screenshot shows the "Generate Test Cases" feature of the Code Refiner tool powered by Generative AI. On the left side, the user has entered a Python function named calculate_discount, which applies different discount rules based on the customer_type and price.

On the right side, the tool has automatically generated relevant unit test cases using the pytest framework. The test cases validate various scenarios, including:

- Regular and member customers with prices above and below 100.
- An edge case with an invalid customer type.

This demonstrates how the tool assists developers in enhancing code reliability by generating robust test coverage with minimal manual effort.

6. CONCLUSION AND FUTURE SCOPE

Conclusion:

AI in code optimization and refactoring can be a basic step in program change building. Hailing from CodeT5, Intel‘s Neural Compressor, Codex, Refactoring Digger, and others, the AI devices have made a difference originators to clarify challenging coding issues, boost perfect execution and undoubtedly optimize code clarity. Such gadgets are engaging progressed progression shapes, managing of specialized commitment, and moving forward the alter of program wanders. In any case, a couple of issues have been recognized that within the occasion that overcome will offer help bring AI to its suitable utilize inside the shown space. Essential challenges impacting reusability are data accessibility, appear transferability, interpretability and extensibility challenges. This ask around shows up that energize enhancement is required of these progresses, making them more viably available, and generalizable over particular coding stages. Inside long-term , creators are set to see more of AI into program headway. Made experiences appear progressives the concept of program making by solidifying highlights such as robotized testing, selfadaptive code, at the side program amalgamation. These progresses are promising to develop engineers, optimize the enhancement handle, and advance program headway get ready proficiently as these progresses progress.

Future Scope:

Looking ahead, the ampleness of fake work range frameworks can be through and through advanced through a few forward-looking methodologies. Firstly, coordination real-time information streams from assembled platforms—such as social media, work sheets, and adaptable apps—will lock in models to memorize and modify to advancing trap strategies viably Other than, joining multimodal analysis—combining substance with pictures, recordings, and without a doubt company logos—can provide wealthier setting and make strides range exactness by recognizing visual signals of off-base posts . Thirdly, leveraging critical learning models, particularly transformer-based models and Bidirectional LSTM, has appeared up guarantee, satisfying precision as tall as 98.7% and AUC scores around 0.91Besides, increasing back for particular tongues and territorial coercion plans guarantees broader show off reasonableness and versatility against around the world duplicity strategies

. At long last, embeddings coherent AI (XAI) strategies like SHAP, LIME, and rule-based considering will not since it were boost client acknowledge but as well offer help accessories in understanding and reacting to making extortion techniques. Together, these progressions position the framework as a cautious, adaptable, and comprehensive secure against persistently progressed fake work postings.

Journal of Information Systems Engineering and Management
2024,9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

Background:

Certain Methodologies of Code Optimization and Refactoring Code optimization and refactoring have been plan concerns in computer program organizing, the foremost reason of which was concerned with making program of higher capability and less inquiring alteration. Insides the past, these assignments were all done by hand, with the objective of a arrange to see at the code and recognize plans of how and where it may require to be optimized. Source code analyzers, which as their title proposes channel through the source code trees without truly executing the code, were among the to start with to support this handle, recognizing such issues as dead code, unused components, and fundamental infringements of coding benchmarks. Lethargic examination, on the other hand, accumulated that the code executed, which progressed creators the chance to require note bothers inside the center of work, for chart, memory spills and bottlenecks. The manual strategies, in appear hate toward of the reality that beneficial for analysing by and colossal arrange programs, some time as of late long laid out or perhaps inefficient especially as the program systems got to be be ceaselessly more essential. In headway, these standard strategies required a essential aggregate of capacity and time to actualize, and in this way being inadequately for twocycle organize. Hence, program building started seeking out for out for the contemplations how this examination and optimization can be wrapped up along these lines with code refactoring being made more fittingly. AI enacted the preeminent discernible move in benchmarks in program orchestrating, progressing approaches and back to engage the mechanization of effortful work. At the to start with steps, robotization in program organizing was exceptionally piddling. A set of scripts and rules for orchestrate work. Be that since it may, with the development of the current time of AI called machine learning (ML), and the another level of AI called basic learning (DL), more able AI applications made, which makes the computer program resistance utilize data to memorize, expect, and in truth allow brilliantly proposal. Insides the development of code optimization and refactoring, AI approaches have been utilized to analyse and anticipate of code for working execution, recognizable organize appearing up inefficient parts, and propose for code alteration of making the code more clear and practical.

REFERENCES

[1]. Panigrahi, "A systematic approach for software refactoring based on class and method level for AI application.," International Journal of Powertrains, , vol. 10, no. 2, pp. pp.143-174., 2021.

[2]. K. a. G. D. Wang, Applying AI techniques to program optimization for parallel computers., 1987.

[3]. Jiang, Supervised machine learning: a brief primer. Behavior therapy, 51(5), pp.675-687., 2020.

[4]. Usama, Unsupervised machine learning for networking: Techniques, applications and research challenges., IEEE access, 7, pp.65579- 65615., 2019.

[5]. A. O. M. Z. N. M. G. a. A. S. Almogahed, Revisiting scenarios of using refactoring techniques to improve software systems

quality., IEEE Access, 11, pp.28800-28819., 2022.

[6]. S. Javaid, "Crowdsourced Data Collection Benefits & Best Practices," 24 oct 2024. [Online]. Available: https://research.aimultiple.com/crowdsourceddata/.

[7]. T. Hospedales, "Meta-Learning in Neural Networks," Samsung AI Center - Cambridge, 2 Sep 2021. [Online]. Available: chat.openai.com/?model=text-davinci-002- render-sha.

[8]. V. C. Vikas Hassija, "Interpreting Black-Box Models: A Review on Explainable Artificial Intelligence," springer links, 24 August 2023. [Online]. Available: https://link.springer.com/article/10.1007/s12559 -023-10179-8.

[9]. S. Kate, "Parallel and Distributed Computing," Medium, 25 April 2023. [Online]. Available: https://medium.com/@sumedhkate/paralleland-distributed-computing-9ee800c9aa8e.

[10]. Y. W. W. J. S. a. H. S. Wang, Codet5: Identifieraware unified pre-trained encoder-decoder models for code understanding and generation., arXiv preprint arXiv:2109.00859., 2021.

[11]. Yue, "CodeT5: The Code-aware EncoderDecoder based Pre-trained Programming Language Models," The 360 Blog, 3 sep 2021. [Online]. Available: https://www.salesforce.com/blog/codet5/.

[12]. M. &. S.-D. A. &. P. I. &. S. S. Sandalski, " Development of a Refactoring Learning Environment. 11.," 2011.

[13]. C. Morrison, "Assessing AI system performance: thinking beyond models to deployment contexts," Microsoft Research Blog, 26 September 2022. [Online]. Available: https://www.microsoft.com/enus/research/blog/assessing-ai-systemperformance-thinking-beyond-models-todeployment-contexts/.

[14]. B. T, "How did I leverage AI and Generative AI in Agile Deployments and in building BizDevOps & DevSecOps pipeline in IT engagements," Linked In, 11 August 2024. [Online]. Available: https://www.linkedin.com/pulse/how-did-ileverage-ai-generative-agile-deploymentsbuilding-balaji-t-iuip