

AI-Enabled SE (AI for SE)

기법/연구 조사

허윤아
26.03.04

Vibe Coding

- Definitions

- 생성된 코드에 대한 최소한의 이해만을 가지고 AI code generation tool에 자연어 prompt로 goal을 제공하여 코딩하는 것 (1)
- 자연어로 된 prompt를 입력하면 AI가 자동으로 code를 생성하고, 이를 사람이 manual testing하는 것 (2)
- AI coding agent가 functional requirements, vibe descriptors, context/constraints를 활용하여 programming하는 것 (3)

- 사용되는 도구들: ChatGPT (4), Replit AI (5), Cursor (6), Claude Code (7), Gemini CLI (8), Codex CLI (9), Windsurf (10) 등

- 기법 (prompt engineering) (2)

- LLM에게 해결해야 할 문제를 설명
 - 무엇을 해결해야 하며, 어떻게 사용자와 상호작용하고, 모듈, 규칙, data flow 단위로 AI를 이해시켜야 함
- 사용할 framework를 선정하고, version을 control하며, AI를 사용하여 debugging
- Context를 풍부하게 제공하여 모호성을 최소화
- Manual code review, testing, IDE compiler 등을 통해서 error를 확인하고 AI agent에 error context를 함께 입력하면 agent가 error fixing을 도움

- 기존 시도들 (youtube 영상)

- 음성을 텍스트로 변환하는 도구와 Cursor를 사용하여 vibe coding 진행 (11)
- Cursor와 browser developer tool을 활용하여 vibe coding과 error message 식별, API call 확인 진행, terminal을 통해 server 쪽 process와 error message monitoring (12)
- Excalidraw를 활용해 계획 및 conceptualization, flow diagram 생성, Code Guide를 통해 project 계획과 문서화 진행 (prompt pre-processing을 통해 hallucination 방지), Cursor를 활용하여 code 생성, project setup과 개발 서버 운영을 위해 terminal 사용, web browser를 통해 testing, verification, 문서 접속 (13)

Code Review

- AutoCommenter: best practice violation을 자동으로 감지하는 code analysis 도구 (14)
 - T5X를 사용하여 Text-to-text transformation, model training 수행
 - Input: task prompt, source code / Target: source code location & URL for best practice violation
 - IDE plugin과 code review system 모두를 통해 AutoCommenter와 상호작용할 수 있음
- AICodeReview: IntelliJ에서 사용 가능한 GPT-3.5 기반의 자동 code review plugin (15)
 - 잠재적 code issue를 식별할 수 있음
 - 다양한 언어 (Java, Kotlin, C++, Python)에 걸쳐 code review 가능
 - 특정 project needs를 만족시키기 위해 code review solution suggestion을 fine-tuning할 수 있음
 - 구체적인 설명과 함께 code revision을 추천
- (16): SW 개발 과정에서 다중 LLM agent를 사용해 code review를 할 수 있는 system
 - Code smell, potential bug, coding standard로부터의 이탈 감지 및 code 개선을 위한 actionable suggestion 제공
 - GPT-4 model을 주로 사용, code review와 bug detection, code smell detection, code optimization 수행 가능
- (17): Agile development에 AI-driven automated code review 적용
 - Agile workflow에 사용되는 CI/CD pipeline에 AI agent를 통합하는 것을 제시

Unit Test Generation

- (18): TestPilot을 통해 JavaScript unit test 자동 생성
 - 25개의 npm packages 내에 있는 API function에 대하여 unit test 생성 및 test 수행
- TestGen-LLM: Android code에 대하여 unit test 자동 생성 (19)
 - 기존 Kotlin test class를 확장하여 corner case cover 및 test coverage 향상에 집중
 - Test equivalence를 자동으로 평가할 수 없으므로 coverage를 proxy로 활용
- AugGPT: 프로그래밍 과제를 위한 unit test 생성에 더 구체적인 prompt와 test quality를 feedback으로 제공 (20)
 - 과제 설명을 context로 하여, syntactic correctness와 semantic depth와 함께 prompt로 제공
 - Statement coverage score와 mutation score를 feedback으로 제공
 - 바라지 않는 test를 걸러내는 rejection strategy와 기대치를 만족하는 unit test가 생성되었을 때 feedback loop을 탈출하고 최종 산출물을 내놓는 exit strategy 설정
- (21): 4종류의 LLM을 활용하여 prompting 성능 비교
 - HumanEval dataset을 unit test code generation의 source로 활용 (LLM evaluation에 사용), 164개의 problems
 - Prompt template을 두 종류로 나누고 docstring 제공 여부를 포함하여 총 세 종류의 prompt format 설정
- (22): 5종류의 LLM을 활용하여 prompting 성능 및 모델 간 성능 비교
 - CodeLlama와 DeepSe다-Coder 기반의 LLM 5종류 활용
 - Prompt 설계의 중요성과 CoT, RAG가 도움이 됨을 보임
- (23): 3종류의 LLM을 활용하여 모델 간 성능 비교
 - Codex, GPT-3.5-Turbo, StarCoder 활용하여 JUnit5 test 생성
 - HumanEval dataset과 47개의 open-source project (SF110) 활용
 - Test coverage, test correctness, test smell, compilation rate 비교

Unit Code Debugging

- (24): 여러 언어를 위한 error detection, optimized code suggestion 도구
 - Web 기반 인터페이스/IDE에 code snippet을 입력하고 프로그래밍 언어를 선택하면 code parsing 진행
 - AI model이 parsed code를 분석하여 syntax, runtime, logical error를 감지
 - Error를 설명하고 이를 수정하기 위한 최적화된 code를 추천
- ChatDBG: LLM이 debugger를 제어할 수 있는 debugging assistant (25)
 - Python, C, C++, Rust 등 언어와 무관하게 사용 가능함
 - 사용자에게 의한 initial prompt를 필요로 하며 이를 통해 LLM이 답해야 하는 내용들과 언제 debugging을 끝낼 것인지를 설정할 수 있음
 - Runtime stack trace를 필요로 하며 이를 통해 root cause를 확인할 수 있음
- (26): 자동 code correction을 위해 LSTM network를 사용하는 Seq2Seq model 제안
 - Tokenization, vocabulary normalization, sequence padding을 포함하는 preprocessing pipeline
 - Erroneous code의 복잡한 pattern을 배우고 정확한 correction을 생성
 - Code sequence가 긴 경우 관련된 section에 집중하도록 하기 위해 attention mechanism 사용
- CodeBlizz: AI-empowered add-on for IDEs (27)
 - Real-time debugging, context-sensitive code suggestion, integrated educational resource 제공
 - CodeBERT, GPT-3와 같은 transformer 기반의 model을 사용하여 code structure를 이해하고 관련 snippet을 생성
 - T5 model을 통해서 real-time error detection과 debugging assistant 수행

References

- 1) Fawzy, Ahmed, Amjed Tahir, and Kelly Blincoe. "Vibe Coding in Practice: Motivations, Challenges, and a Future Outlook--a Grey Literature Review." arXiv preprint arXiv:2510.00328 (2025).
- 2) Sapkota, Ranjan, Konstantinos I. Roumeliotis, and Manoj Karkee. "Vibe coding vs. agentic coding: Fundamentals and practical implications of agentic ai." arXiv preprint arXiv:2505.19443 (2025).
- 3) Bamil, Vinay. "Vibe Coding: Toward an AI-Native Paradigm for Semantic and Intent-Driven Programming." arXiv preprint arXiv:2510.17842 (2025).
- 4) J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altschmidt, S. Altman, S. Anadkat, et al., "Gpt-4 technical report," arXiv preprint arXiv:2303.08774 (2023).
- 5) Replit AI, <https://replit.com/ai>
- 6) Cursor, <https://cursor.com/en-US>
- 7) Claude, <https://claude.ai/>
- 8) Gemini CLI, <https://geminicli.com/>
- 9) Codex CLI, <https://openai.com/ko-KR/codex/>
- 10) Windsurf, <https://windsurf.com/>
- 11) VIBE CODING with CURSOR – Draw w/ HTML and JS, <https://www.youtube.com/watch?v=irBnUAq3MAw>
- 12) Vibe Coding with AI: Watch me fix a chat interface in real-time in this long video, <https://www.youtube.com/watch?v=EAPWrpvIOjs>

References

- 13) Vibe Coding Startup Using Most Advanced AI— Phase 1, <https://www.youtube.com/watch?v=SNARzs6jzWY>
- 14) Vijayvergiya, Manushree, et al. "Ai-assisted assessment of coding practices in modern code review." Proceedings of the 1st ACM international conference on AI-powered software. 2024.
- 15) Almeida, Yonatha, et al. "AIcodeReview: Advancing code quality with AI-enhanced reviews." SoftwareX 26 (2024): 101677.
- 16) Rasheed, Zeeshan, et al. "Ai-powered code review with llms: Early results." arXiv preprint arXiv:2404.18496 (2024).
- 17) Ahmed, Saad. "Integrating AI-driven automated code review in Agile development: Benefits, challenges, and best practices." International Journal of Advanced Engineering, Management and Science 11.2 (2025): 1-10.
- 18) Schäfer, Max, et al. "An empirical evaluation of using large language models for automated unit test generation." IEEE Transactions on Software Engineering 50.1 (2023): 85-105.
- 19) Alshahwan, Nadia, et al. "Automated unit test improvement using large language models at meta." Companion Proceedings of the 32nd ACM International Conference on the Foundations of Software Engineering. 2024.
- 20) Zheng, Kaisheng, Yuanyang Shen, and Yida Tao. "Automatic Unit Test Generation for Programming Assignments Using Large Language Models." 2025 IEEE/ACM 37th International Conference on Software Engineering Education and Training (CSEE&T). IEEE, 2025.

References

- 21) Lira, Werney Ayala Luz, et al. "Evaluating the Effectiveness and Cost-Efficiency of Large Language Models in Automated Unit Test Generation." Simpósio Brasileiro de Qualidade de Software (SBQS). SBC, 2025.
- 22) Yang, Lin, et al. "On the evaluation of large language models in unit test generation." Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering. 2024.
- 23) Siddiq, Mohammed Latif, et al. "Using large language models to generate junit tests: An empirical study." Proceedings of the 28th international conference on evaluation and assessment in software engineering. 2024.
- 24) Varshini, Polani Keerthi, et al. "AI-POWERED CODE DEBUGGING ASSISTANT." International Journal of Marketing Management 13.4 (2025): 84-91.
- 25) Levin, Kyla H., et al. "Chatdbg: Augmenting debugging with large language models." Proceedings of the ACM on Software Engineering 2.FSE (2025): 1892-1913.
- 26) Khyzer, H., R. M. Rashid, and A. Khizer. "Human–AI collaboration in data-driven software engineering: A Seq2Seq model with attention for automated code debugging." International Journal of Sciences and Emerging Technologies 3.1 (2025).
- 27) Preethi, P., et al. "Towards codeblizz: developing an ai-driven ide plugin for real-time code suggestions, debugging, and learning assistance with generative ai and machine learning models." 2024 International Conference on Emerging Research in Computational Science (ICERCS). IEEE, 2024.
- 28) Coding with ChatGPT and Other LLMs: Navigate LLMs for effective coding, debugging, and AI-driven development (책. 읽어보지는 못했으나 도움이 될 것으로 예상)