

SW 개발자 역량 평가를 위한 AI 에이전트에 관한 연구

김준형¹, 송재훈², 윤정훈³, 이곤우⁴, 조재용⁵, 정명훈⁶

¹⁻⁵ 경북대학교 컴퓨터학부 학부생

⁶ 구글코리아

junhyung85920@gmail.com, jhssong02@gmail.com, junghoonyun@knu.ac.kr, ery0508@knu.ac.kr,
researcherdragon@gmail.com, javalove93@gmail.com

A Study on an AI Agent for Assessing SW Developer Competency

Jun-Hyeong Kim¹, Jae-Hoon Song², Jeong-Hoon Yun³, Gon-Woo Lee⁴, Jae-Yong Cho⁵,
Myeong-Hoon Jeong⁶

¹⁻⁵ Dept. of Computer Science, Kyung-pook National University

⁶ Google Korea

요약

본 논문은 GitHub 데이터와 설문을 활용한 AI Agent[1]기반의 SW 개발자 역량 평가 시스템을 제안한다. Chain of Thought 기법이 적용된 AI Agent 와 합당한 평가 기준을 바탕으로 체계적인 평가를 제공한다. 평가 프로세스의 정밀도와 신뢰성을 높이고, AI 기반의 답변 검증 과정을 도입하여 평가 결과의 객관성을 확보할 수 있다는 점을 시사한다.

1. 서론

최근 IT 산업의 급속한 발전과 함께 소프트웨어 개발에 대한 수요가 기하급수적으로 증가하고 있다.[2] 이에 따라 기업은 SW 개발자의 기술력과 문제 해결 능력을 객관적으로 평가할 필요성이 커졌다. 전통적인 평가 방식은 평가자의 주관이 개입될 가능성이 높고, 평가 기준의 일관성을 확보하기 어렵다는 한계가 있다. 특히, 전문 인력의 부족과 업무 부담으로 인해 중소규모 기업에서는 SW 개발자 역량을 체계적으로 평가하는 것이 큰 도전 과제로 대두되고 있다. 이러한 문제점을 해결하기 위해, 본 연구에서는 AI Agent 를 활용한 자동화된 SW 개발자 역량 평가 시스템을 구현하고자 한다. 최근 대형 언어 모델(LLM) 사용 비용이 줄어들고 있어, AI Agent 기반의 자동화된 평가 시스템 도입에 대한 경제적 부담이 줄어들고 있다는 점도 고려할 만하다. 이 시스템은 SW 개발자의 Github 활동 데이터와 협업 지표 등 다양한 정보를 분석하여 평가의 객관성과 신뢰성을 확보하는 것을 목표로 한다.

2. 시스템 구조 및 AI Agent 설계

본 시스템은 Web, Bot을 비롯한 UI와 AI Agent, 그리고 Agent가 사용할 GitHub API, Embedding API, LLM API, Vector DB로 구성된다. AI Agent가 프롬프트를 처리하는 과정은 다음과 같다. 에이전트는 사용자로

부터 전달받은 요청을 여러 단계의 프롬프트로 분해한 후, 각 단계에 필요한 정보(RAG 데이터셋)를 다양한 도구를 통해 수집한다. 이후 대형 언어 모델(LLM)을 이용하여 산출된 답변의 적합성을 검증하는 과정을 거치며, 이러한 Chain of Thought 기법을 통해 SW 개발자의 역량을 정밀하게 평가한다.

3. AI Agent의 역량 평가 방법

본 연구에서는 소프트웨어 개발자의 역량을 <표 1>과 같이 네 가지 항목으로 구분하여 정량적으로 평가한다. 각 항목은 GitHub 활동 데이터를 기반으로 분석되며, 개별 단위 (이슈, 커밋, PR, 리뷰 등)에 점수를 부여한 뒤 정규화하여 0~100 점 범위로 환산한다. 이를 통해 정량적 이면서도 비교 가능한 평가 점수를 산출할 수 있다.

3.1 문제 해결 (Problem Solving)

문제 해결 항목은 개발자가 해결한 GitHub 이슈 각각에 대해 상대적인 난이도 점수를 부여하는 방식으로 평가한다. 각 이슈는 해결의 복잡도, 기술 난이도, 커뮤니케이션을 포함한 해결 과정 등을 고려하여 점수가 부여된다. 모든 이슈의 점수 집합을 s_1, s_2, \dots, s_n 이라 할 때, 문제 해결 항목의 정규화된 점수 H_1 은 다음과 같이 계산된다:

$$H_1 = \left(\frac{1}{n} \sum_{i=1}^n s_i / \max_{1 \leq i \leq n} s_i \right) \times 100$$

3.2 구현 기능 (Implemented Functionality)

구현 기능 항목은 커밋 메시지를 기반으로 기능의 구조적 중요도와 시스템에 미친 영향력을 분석하여 점수를 부여한다. 커밋 메시지의 키워드를 기반으로 각 커밋의 상대적 중요도를 산출하며, 이를 점수화한 후 다음과 같이 정규화한다:

$$H_2 = \left(\frac{1}{n} \sum_{i=1}^n s_i / \max_{1 \leq i \leq n} s_i \right) \times 100$$

3.3 활동성 (Activity)

활동성은 개발자가 일정 기간 동안 수행한 커밋 수(c)와 Pull Request 수(p)를 기반으로 평가한다. 기준 커밋 수(C_{\max}) 및 기준 PR 수(P_{\max})는 평가 기간과 조직 또는 커뮤니티의 평균 활동 데이터를 기반으로 설정된다. 커밋과 PR의 중요도는 각각 $\alpha = 0.5$, $\beta = 0.5$ 의 가중치를 부여하며, 활동성 점수 H_3 는 다음과 같이 산출한다:

$$H_3 = \left(\frac{c}{C_{\max}} \cdot \alpha + \frac{p}{P_{\max}} \cdot \beta \right) \times 100$$

3.4 협업 능력 (Communication & Collaboration)

협업 항목은 GitHub의 코드 리뷰 응답 시간을 기반으로 평가하며, 빠른 응답일수록 높은 점수를 부여한다. 평균 리뷰 응답 시간을 \bar{r} , 최대 허용 응답 시간을 r_{\max} 라 할 때, 협업 항목의 점수 H_4 는 다음과 같이 계산된다:

$$H_4 = \left(1 - \frac{\bar{r}}{r_{\max}} \right) \times 100$$

<표 1> 평가 항목별 평가 자료

| 평가 항목 | 평가 자료 |
|-------|--|
| 문제 해결 | GitHub 저장소의 이슈(issue)의 난이도 및 개수 |
| 구현 기능 | GitHub 저장소의 커밋(commit) 내용 |
| 활동성 | GitHub 저장소의 커밋(commit) 및 풀 리퀘스트(pull request) 개수 |
| 협업 능력 | GitHub 저장소의 평균 리뷰 응답시간, 평균 리뷰 시간 |

3.5 종합 역량 점수 산출

각 항목별 평가 단위 수(이슈 수, 커밋 수 등)를 n_j 라 하고, 총 항목 수를 m 이라 할 때, 항목별 가중치 w_j 는 다음과 같이 설정한다:

$$w_j = \frac{n_j}{\sum_{k=1}^m n_k}$$

최종 역량 점수 F 는 항목별 정규화 점수 H_j 에 대해 가중 평균을 취하여 다음과 같이 계산된다:

이 방식은 각 항목의 평가 규모에 비례한 공정한 가중치를 자동 부여한다. 이를 통해 AI 기반 역량 평가의 객관성과 신뢰성을 향상시킬 수 있다.

$$\text{최종 점수 } (F) = \sum_{j=1}^m w_j \cdot H_j$$

4. 한계

이 시스템의 한계로는 두 가지를 들 수 있다. 먼저, SW 개발자 역량 평가에 필요한 데이터가 부족함에 더하여 정량화하기 어려운 비정형 역량 데이터를 완벽히 반영하기 어렵다. LLM에게 평가 요청 시 필요한 점수에 대한 객관적인 기준을 모색해야 한다. 다음으로 각 평가 항목의 가중치 설정 및 알고리즘의 신뢰성에 대한 추가 검증이 필요하며, 다양한 개발 환경과 산업 특수성을 고려한 보완 연구가 요구된다.

5. 결론

본 논문에서는 국내 소프트웨어 시장의 성장과 이에 따른 SW 개발자 역량 평가의 필요성을 바탕으로, AI Agent를 활용한 SW 개발자 역량 평가 시스템의 설계 및 구현 방안을 제시하였다. 전문 인력의 부족으로 인해 역량 평가에 어려움을 겪는 환경에서, AI Agent 기반의 자동화된 평가의 객관성과 정밀도를 확보할 수 있음을 시사한다. 향후 연구에서는 다양한 데이터 소스의 확장과 평가 시스템의 정교화를 통해 정확도를 더욱 높이고, 실무 환경에서의 적용 가능성을 검증하고자 한다. 이를 통해 IT 산업 전반의 인재 관리 및 역량 향상에 기여할 수 있을 것으로 기대된다.

ACKNOWLEDGEMENT

“이 연구는 과학기술정보통신부 및 정보통신기획평가원의 SW 중심대학사업 지원을 통해 수행되었음” (2021-0-01082)

참고문헌

- [1] Tula Masterman, Sandi Besen, Mason Sawtell, Alex Chao. "The Landscape of Emerging AI Agent Architectures for Reasoning, Planning, and Tool Calling: A Survey" arXiv:2404.11584
- [2] SPRi. 2023년 소프트웨어산업 연간보고서. 서울, 소프트웨어정책연구소, 2024
- [3] Forsgren Nicole, Storey Margaret-Anne, Maddila Chandra, Zimmermann Thomas, Houck Brian, Butler Jenna. "The SPACE of Developer Productivity: There's more to it than you think." ACM Transactions on Software Engineering and Methodology, 42(2), 13-17, 2021