JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 30 No. 10, pp. 11-22, October 2025
https://doi.org/10.9708/jksci.2025.30.10.011

A Generative AI-Based Personalized Programming Education System with Adaptive Rubric Evaluation

Euhee Kim*

*Professor, Dept. of Software Convergence, Shinhan University, Gyeonggi-do, Korea

[Abstract]

This paper presents a personalized programming education system that integrates a GPT-4-based role-play simulation and an adaptive rubric-driven automated evaluation engine. The system employs prompt engineering to guide GPT-4 in dual roles—as a conversational tutor and as an evaluator—enabling dynamic learner modeling and self-explanation through contextual dialogue. A total of 450 question–answer interactions were collected across three instructional scenarios and three learner personas. Experimental results demonstrated improvements in learning outcomes through repeated sessions, with rubric score consistency within ±0.5 points and a 91.3% agreement rate between AI-generated and human evaluator scores. These findings confirm the system's effectiveness in delivering real-time personalized feedback and formative assessment, and suggest practical applications for AI-driven education.

▸ **Key words**: generative AI, prompt engineering, programming education, adaptive rubric evaluation, role-play simulation, self-explanation strategy, automated feedback

[요 약]

본 논문은 GPT-4 기반의 역할극 시뮬레이션과 적응형 루브릭 기반 자동 평가 모듈을 통합한 맞춤형 프로그래밍 교육 시스템을 제안한다. 본 시스템은 GPT-4가 대화형 튜터와 평가자의 이중 역할을 수행하도록 프롬프트 엔지니어링을 활용하여, 문맥 중심의 대화를 통한 학습자 모델링과 자기설명을 유도한다. 실험은 3가지 교육 시나리오와 3단계 학습자 페르소나를 조합하여 총 450개의 질의–응답 데이터를 수집하였다. 반복 세션을 통해 학습 성과 향상이 확인되었고, 루브릭 점수는 ±0.5 이내의 일관성을 보였으며, 사람 평가자와 91.3%의 평가 일치율을 나타냈다. 이 결과는 본 시스템이 실시간 개인 맞춤형 피드백과 구조화된 형성 평가를 효과적으로 제공할 수 있음을 실증하며, AI 기반 교육의 실용 가능성을 제시한다.

▸ **주제어**: 생성형 AI, 프롬프트 엔지니어링, 프로그래밍 교육, 적응형 루브릭 평가, 학습자 모델링, 대화형 튜터, 학습자 페르소나, 역할극 시뮬레이션, 자기 설명 전략, 자동 피드백

• First Author: Euhee Kim, Corresponding Author: Euhee Kim
*Euhee Kim (euhkim@shinhan.ac.kr), Dept. of Software Convergence, Shinhan University
• Received: 2025. 09. 05, Revised: 2025. 09. 30, Accepted: 2025. 10. 09.

http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

I. Introduction

프로그래밍 교육은 반복 학습과 피드백 중심의 학습 전략이 핵심적인 역할을 하는 분야이다. 특히 초보 학습자들은 개념적 오개념, 문법 오류, 논리적 구조의 불완전성 등 복합적인 인지적 장애를 겪기 쉬우며, 이러한 오류를 인식하고 수정하는 과정에서 피드백이 학습 효과에 결정적인 영향을 미친다. 그러나 현재 대부분의 온라인 코딩 학습 플랫폼은 정답 여부만을 판별하는 자동 채점 시스템에 기반하고 있어, 학습자의 사고 흐름이나 오개념 유형을 분석하거나 그에 맞는 정성적 피드백을 제공하는 데 한계가 있다. 이러한 환경에서는 학습자가 스스로 사고를 전개하거나 오류 원인을 탐색하는 능력을 충분히 기르기 어렵다[1-3].

이러한 문제를 해결하기 위한 대안으로, 최근 GPT-4와 같은 대규모 언어 모델(Large Language Models, LLMs)의 등장은 자연어 기반 상호작용을 가능하게 하며, 시나리오 기반의 역할극(role-play) 구조를 활용한 교육적 접근에 관한 관심을 높이고 있다. 생성형 AI를 교수자, 면접관, 상담자 등으로 설정하여 학습자의 자기설명(self-explanation), 몰입도, 자율성 등을 향상하게 시키려는 연구가 진행되고 있으며, 이는 반복 설명, 맞춤 피드백, 학습자 수준별 대응 등을 자동화함으로써 인간 교사가 제공하기 어려운 교육적 서비스를 보완할 수 있다는 점에서 새로운 교육 기술의 전환점을 제시하고 있다[4-5].

특히 언어 교육 분야에서는 GPT-4 기반 AI 피드백 시스템이 초급 한국어 학습자의 문법 및 어휘 오류에 대해 자동 피드백을 제공하며, 반복 학습과 자기주도 학습을 유도하는 효과가 확인된 바 있다[6]. 해당 연구는 AI 기반 자동 평가의 신뢰성과 타당성을 검증하기 위해 사람 평가자와의 루브릭(rubric) 점수 일치율을 정량적으로 분석하였으며, 그 결과 AI 평가가 일정 수준 이상의 일관성과 타당성을 갖춘 것으로 나타났다. 다만, 언어 교육의 문맥에서는 AI의 이해 한계와 개별화 피드백의 부족이 한계로 지적되었으며, 이는 향후 역할극 구조와 페르소나(persona) 기반 설정을 통한 보완이 요구된다.

프로그래밍 교육 분야에서도 생성형 AI를 활용한 시뮬레이션 연구가 시도되었다. 대표적으로 Python 학습 트리 기반 구조를 활용하여 학습자의 디버깅 사고 흐름을 시뮬레이션하고, 다양한 코드 수정 경로를 생성하는 방식으로 개인화된 학습 전략을 구현한 연구가 있다[7-8]. 이 연구는 LLM을 활용해 학습자의 오류 탐색 행동을 구조적으로 재현했다는 점에서 의의가 있으나, 정형화된 답변 흐름에 기반하여 자유로운 질의응답, 학습자 수준별 페르소나 구성, 루브릭 기반 자동 평가 등의 측면에서는 한계가 존재한다.

이에 본 연구는 기존 연구의 한계를 보완하고자, 생성형 AI의 고도화된 언어 처리 능력과 프롬프트 엔지니어링(prompt engineering) 기법을 활용하여, 학습자와 AI 튜터 간의 시나리오 기반 자유 질의응답이 가능한 역할극 시뮬레이션 시스템을 설계·구현하였다. 아울러, AI 튜터의 응답을 교육적 기준에 따라 정량적으로 평가할 수 있도록, 학습자의 수준과 과제 유형에 따라 평가 항목과 가중치를 동적으로 조정하는 '적응형 루브릭 생성기'를 통합한 자동 평가 모듈도 함께 제안하였다.

제안된 시스템은 반복 학습 세션을 통해 학습자의 질문 구조와 AI 응답 품질의 변화를 분석하며, 응답의 정확성, 일관성, 비환각성(hallucination-free)을 검토하고, 루브릭 점수 및 피드백을 실시간으로 제공함으로써 학습자의 자기설명 전략을 유도한다. 이를 통해 기존의 정답 중심 채점 방식에서 벗어나, 생성형 AI 기반의 역할극-자동 평가 통합 구조가 프로그래밍 교육 현장에서 실질적인 피드백 제공, 학습자 중심 설계, 반복 기반 학습 효과에 이바지할 수 있음을 실증적으로 제시하고자 한다. 단, 시스템 실험은 가상의 학습자 페르소나를 중심으로 시뮬레이션 실험을 진행하였기에, 실제 학습자를 대상으로 한 장기적 검증이 필요하다.

II. Related works

본 장에서는 최근 생성형 AI 기반 교육 응용 연구의 동향을 살펴보고, 기존 튜터링 시스템과 비교하여 시나리오 다양성, 루브릭 정량화, 반복 피드백 구조 및 평가 프롬프트의 적응형 자동 생성 시스템으로의 확장 가능성을 갖는지를 설명한다.

2.1 Generative AI in Education

LLM을 기반으로 한 생성형 AI는 최근 교육 분야에서 다양한 시도로 확장되고 있다. 특히 GPT-3, GPT-4 등의 모델은 자연어 이해 및 생성 능력을 바탕으로 대화형 튜터, 자동 요약기, 코드 생성기 등의 형태로 활용되고 있으며, 학습자와의 자유로운 상호작용을 통해 실시간 피드백을 제공할 수 있는 장점이 있다[9].

Chu et al.은 LLM 기반 에이전트가 교육 분야에서 수행하는 역할을 교사·학생을 지원하는 교수설계 에이전트와

특정 분야에 특화된 교육 에이전트의 두 가지로 구분하여 이러한 에이전트들의 기술적 기반과 프라이버시, 편향, 환각, 시스템 통합 등을 조사 분석하였다. 이를 통해 교육 현장에서 LLM 에이전트의 활용 가능성과 향후 연구 협력의 방향성을 제시하였다[4].

Mollick et al.은 생성형 AI 에이전트를 활용해 개인화된 교육 시뮬레이션을 스타트업 피치 코칭, 면접 연습 등의 역할을 수행하면서 학습자에게 비판적 피드백을 제공하는 시스템을 구축하였고, 학습자의 몰입도와 실습 효과가 유의하게 향상됨을 보고하였다[5].

또한, Ross et al.은 C 프로그래밍 교육을 위한 LLM 기반 튜터인 GuideLM을 개발하고, 교육 효과를 높이기 위해 ChatGPT-4o에 감독 학습(supervised learning) 방식으로 기존 모델 대비 소크라테스식 질문 유도와 간결한 설명에서 우수한 성과를 보여주었다[10].

이와 같은 선행 연구들은 생성형 AI가 교육 대화나 학습 시뮬레이션 도구로 활용될 수 있음을 보여주었으나, 대부분 단일 회차 기반의 응답 생성에 초점을 맞추고 있으며, 학습자의 능동적인 사고 흐름이나 시나리오 전개에 따라 행동을 조정하는 구조는 부족함을 보여주었다.

프로그래밍 교육에서 학습자 시뮬레이션은 핵심적인 연구 영역으로, 특히 학습자의 인지 과정과 문제 해결 전략을 정밀하게 모사하는 것이 효과적인 교육 시스템 설계에 중요하다. 이와 관련하여 최근 Wei et al.은 '사고의 연결 고리(chain-of-thought, CoT)' 프롬프트가 대규모 언어 모델의 복잡한 추론 능력을 유의미하게 향상하게 한다는 점을 실험적으로 입증하였다[7]. CoT 프롬프트는 언어 모델이 연속적인 사고 단계를 명시적으로 표현하도록 유도함으로써, 문제 해결 과정을 학습자와 유사한 방식으로 시뮬레이션할 수 있도록 한다.

이러한 CoT 방식은 단순한 정답 예측을 넘어, 학습자의 사고 흐름과 자기설명 전략을 모사할 수 있다는 점에서 학습자 시뮬레이션 설계에 직접적인 시사점을 제공한다. 실제로 Zhan et al.은 CoT 프롬프트 기법을 Python 학습 트리(Programming Tree of Thought, PToT)에 확장 적용하여, 코딩 행위, 오류 유형, 반응 패턴 등을 시뮬레이션하는 CoderAgent 시스템을 제안하였다[8]. 이 시스템은 AI 튜터가 시뮬레이션 된 학습자의 반응을 바탕으로 맞춤형 피드백을 제공하는 구조를 갖추고 있지만, 루브릭 기반의 정량 평가 기능은 포함되어 있지 않으며, 학습자가 자유롭게 대화를 주도하기보다는 사전에 정해진 학습 경로 내에서 제한적으로 작동한다는 점에서 한계가 존재한다.

2.2 Automated Assessment & Rubric Systems

교육 현장에서 루브릭 평가는 학습자의 성취를 다면적으로 진단하고 교수자 간 평가의 일관성을 확보하기 위한 중요한 도구로 널리 활용되고 있다. 최근에는 이러한 루브릭 평가 기준을 생성형 AI가 학습하거나 이를 기반으로 자동 평가를 수행하도록 설계하려는 시도들이 활발히 이루어지고 있다. 이와 관련하여 Wang et al.은 다양한 유형의 Python 코드 생성 문제에 대해 프롬프트 엔지니어링을 활용하여 평가 기준을 명확히 제시함으로써, LLM의 응답 안정성을 향상했으며, 이후 생성된 결과물을 AI 스스로 평가하도록 구성한 구조를 실험하였다. 이 연구는 컴퓨터 프로그래밍 교육에서 다양한 교육 목적에 맞춘 프롬프트 전략을 체계적으로 설계하고 평가함으로써, 학습자의 수준에 따라 성과를 향상하는 데 있어 GPT-4와 같은 모델이 효과적일 수 있음을 입증하였다[11-12].

또한 Phung et al.은 학생이 작성한 피드백 데이터를 활용해 생성형 AI 모델을 파인튜닝(fine tuning)함으로써, 보다 인간 중심적이고 공감 기반의 자동 피드백 시스템을 제안하였으며, C 프로그래밍 과제를 대상으로 자동 채점과 피드백 제공의 효과를 실험하였다[13].

Kasneci et al.은 ChatGPT의 응답 품질을 사람 평가자와 비교 분석하여, 일부 루브릭 항목에서는 인간과 유사한 판단을 내릴 수 있다는 점을 보고하였다[14].

이와 같은 선행 연구들은 생성형 AI가 교육적 평가 도구로 활용될 가능성을 보여주지만, 대부분이 학습자의 질의 흐름이나 인지 과정과는 무관하게 정적인 기준만을 적용하여 평가를 수행하거나, 평가 결과가 개별화된 피드백으로 이어지지 않는다는 한계를 지닌다. 또한 학습 시나리오 전반에서 평가와 피드백이 통합적으로 작동하지 않아, 실시간 교육 맥락에 최적화된 자동 평가 시스템으로 발전하기에는 다소 제한적인 측면이 존재한다.

2.3 Personalized Conversational Tutor System

LLM의 발전은 교육 분야에서 실시간 상호작용 기반의 대화형 학습 지원 시스템으로의 확장 가능성을 제시하고 있다. 특히 GPT-4와 같은 최신 LLM은 프롬프트 엔지니어링 기법을 통해 인간의 추론 방식을 모방하고, zero-shot(사전 예시 없이 문제 해결) 또는 few-shot(적은 수의 예시로 문제 해결) 기반 CoT 추론을 활용하여 복잡한 학습 질의에 대해 단계적인 사고 흐름을 생성할 수 있다.

이와 같은 구조를 교육에 적용한 대표적인 사례로 Park et al.은 GPT-4를 기반으로 한 개인 맞춤형 대화형 튜터링 시스템을 제안하였다[15]. 해당 시스템은 학습자의 이

해 수준, 오류 유형, 학습 진행 상황 등 다양한 정보를 반영하는 다면적 학습자 모형화를 구성하고, 이를 반영한 프롬프트 템플릿(base prompt + personalized prompt) 구조를 통해 LLM이 개인화된 응답을 생성하도록 설계되었다. 여기서 base prompt는 역할, 시나리오 목적 등 공통 프레임을 제시하고, personalized prompt는 학습자의 프로파일, 피드백 기록, 감정 상태 등을 동적으로 반영하여 응답의 적합성과 맥락성을 강화한다. 또한, 학습 세션 내 질의-응답을 반복하면서 학습자의 상태 변화는 프롬프트 업데이트에 반영되고, 세션 종료 후에는 LLM이 학습자에게 맞춤형 요약 및 추천을 제공함으로써 CoT 기반 자기주도 학습 흐름이 완성된다. 다만 이 시스템은 특정 영어 작문 과목에 국한되며, 응답에 대한 정량적 루브릭 평가 체계가 내장되어 있지 않아 학습 효과의 구조적 검증에는 한계가 있다.

이에 반해 본 연구에서는 GPT-4 기반 시나리오-페르소나 구조를 통해 학습자 수준에 적합한 질의-응답 흐름을 구현하고, AI 응답에 대해 자동 루브릭 평가와 피드백 생성을 통합한 구조화된 시스템을 제안한다. 특히 '적응형 루브릭 생성기'를 통해 학습자의 수준과 과제 유형에 맞게 평가 항목과 가중치를 동적으로 설정함으로써, 평가의 유연성과 타당성을 확보하였다. 이는 기존의 진단 기반 대화형 튜터 시스템의 한계를 보완하고, 프롬프트 구조 설계-CoT 유도-자동 평가-실시간 피드백을 유기적으로 연결한 실용적 학습 분석 프레임워크를 제시하는 데 의의가 있다.

III. Methodology

이 장에서는 GPT-4 기반 생성형 AI을 활용하여 학습자와 AI 튜터 간의 실시간 상호작용이 가능한 역할극 기반 시뮬레이션 학습 환경을 구현하고, 이에 따른 AI 응답을 정량적으로 평가할 수 있는 루브릭 기반 자동 평가 엔진을 통합한 프로그래밍 교육 프레임워크를 제안한다. 전체 시스템은 대화형 시뮬레이터, 학습자 페르소나 설계, 자동 루브릭 평가 엔진의 세 가지 주요 구성 요소로 이루어져 있으며, Streamlit 기반 웹 애플리케이션 형태로 구현되었다.

3.1 Role-Play-Based Conversational Simulator

본 시스템의 중심에는 GPT-4 API를 활용한 대화형 생성형 AI 기반 시뮬레이터가 존재한다. 이 시뮬레이터는 역할극 개념을 적용하여, AI가 특정 역할을 수행하며 사용자의 자연어 질의에 응답하는 구조를 갖는다. 여기서 역할극은 교육적 맥락에서 특정 교수자 역할을 수행하도록 인공지능에게 사전 정의된 행동 원칙과 응답 전략을 부여하여, 학습자의 수준과 시나리오 목적에 맞춘 상호작용을 유도하는 방식이다.

시뮬레이터는 사전에 설계된 역할 프롬프트 템플릿을 바탕으로 세 가지 시나리오— Python 문법 튜터, Pandas 데이터 분석 조교, 객체지향 프로그래밍 개념 설명자—에서 각기 페르소나 설정에 따라 작동한다. 각 역할은 학습자의 수준에 적합한 언어 표현, 설명 방식, 오류 지적 전략, 코드 예시 활용 등을 다르게 구성함으로써 실시간 대화 기반의 튜터링을 제공한다.

특히, 시뮬레이터는 반복 세션 내에서 학습자의 질의 유형, 피드백 반응, 오류 패턴을 누적 반영하여, 응답의 맥락을 점진적으로 조정하고 발전시키는 기능을 갖는다. 이로써 단순 질의응답을 넘어서 실제 교육 현장에서의 상호작용과 유사한 몰입도 높은 학습 경험을 가능하게 한다.

3.2 Learner Persona Design

역할극 기반 프로그래밍 교육 시뮬레이터의 교육적 효과를 극대화하기 위해, 본 연구에서는 실제 프로그래밍 교육 환경에서 자주 관찰되는 학습자의 특성과 오류 유형을 반영하여 세 가지 수준의 학습자 페르소나(초급, 중급, 고급)를 설계하였다. 페르소나는 단순한 정답 여부 판단을 넘어, 학습자의 수준과 질문 유형, 오류 발생 양상, 피드백 요구 방식 등 다양한 상호작용 요소를 반영한다.

특히 본 연구에서는 공통된 수준 구분 체계를 기반으로 하되, 각 시나리오의 특성에 따라 학습자의 오류 유형과 질문 양식이 달라지도록 설계하였다. 예를 들어, 초급 수준 학습자의 경우 Python 문법 시나리오에서는 변수 선언 오류, Pandas 시나리오에서는 인덱싱 혼동, OOP 시나리오에서는 클래스와 인스턴스 개념 구분 실패가 주로 나타난다. 이러한 페르소나 기반 차별화 설계는 GPT-4가 시나리오 맥락과 학습자 상태를 종합적으로 고려한 응답을 생성할 수 있도록 하며, 실제 교수자의 대응 전략을 근사화하는 데 이바지한다.

또한, 반복 세션에서 학습자의 질의가 점차 구체화하고, AI 응답이 학습자 행동에 맞추어 적응하여 조정됨에 따라, 본 시스템은 자기설명 기반 학습을 촉진하는 역할을 수행한다.

3.3 AI Rubric Feedback Evaluation Engine

GPT-4 기반 시뮬레이터가 학습자의 자연어 질의에 대해 피드백을 생성한 이후, 본 시스템은 동일한 GPT-4 모

델을 평가자 역할로 재설정하여 자동 평가 프로세스를 수행한다. 이 평가 엔진은 사전에 정의된 루브릭 항목을 기준으로 응답을 채점하며, 각 항목은 0~2점으로 점수화되어 총점은 10점 만점이다. 점수화뿐 아니라, 항목별 채점의 근거와 개선을 위한 제안을 자연어 형태로 피드백 생성함으로써, 학습자가 자신의 사고 흐름을 점검하고 자기 설명을 수행할 수 있는 구체적인 학습 단서를 제공한다.

예를 들어 "Python 문법" 시나리오에서 생성된 AI 피드백은 Table 1에서 제시된 루브릭 평가 기준 항목에 따라 자동 평가된다. 각 항목은 AI 응답의 교육적 완성도와 피드백 품질을 동시에 평가하도록 설계되었으며, 이 피드백은 Streamlit 기반 웹 인터페이스를 통해 실시간으로 사용자에게 반환된다. 사용자는 점수와 함께 제공되는 자연어 설명을 바탕으로 이해의 정도를 점검하고, 반복 학습을 통해 점차 명확하고 정돈된 질문을 생성할 수 있다.

Table 1. A Rubric-based Evaluation for Python Tutor

Evaluation Criteria	Score Range	Evaluation Description
Accuracy	0-2 pts	Logical soundness of concept explanation and code
Relevance	0-2 pts	Alignment with learner's question
Clarity	0-2 pts	Step-by-step explanation and ease of understanding
Use of Examples	0-2 pts	Inclusion of real code or case examples
Logical Flow	0-2 pts	Coherence and consistency in explanation structure and flow

3.4 Streamlit-Based User Web Interface

앞선 3.3절에서 설명한 루브릭 기반 자동 평가 엔진은 실시간 피드백 제공이라는 교육적 효과를 극대화하기 위해, 직관적이고 반응성이 높은 사용자 인터페이스와 통합되어야 한다. 이를 위해 본 연구는 Python 기반의 Streamlit 웹 프레임워크를 활용하여 전체 시스템을 웹 앱 형태로 구현하였다.

해당 인터페이스는 사용자가 별도의 설치 없이 웹 브라우저에서 직접 접근할 수 있으며, 역할극 시나리오 선택부터 자연어 질의 입력, GPT-4 응답 확인, 자동 루브릭 평가 점수, 대화 이력과 피드백 확인까지 일련의 과정을 단일 화면 내에서 대화형으로 수행할 수 있도록 설계되었다.

Fig. 1. Streamlit-based User Web Interface

예를 들어, Fig. 1는 본 시스템의 실제 웹 인터페이스 구성을 시각화한 예시이다. 사용자는 먼저 학습 시나리오와 자신의 학습 수준을 선택하고, 자유롭게 질문 또는 코드 조각을 자연어로 입력한다. GPT-4는 선택된 시나리오와 페르소나 프롬프트를 기반으로 튜터 역할을 수행하며, 이에 대한 응답을 출력한다. 동시에 동일한 GPT-4가 평가자 역할로 자동 전환되어 응답에 대한 루브릭 평가 점수와 자연어 피드백을 자동 생성하여 사용자에게 실시간으로 반환한다. 이 모든 과정은 Streamlit 상에서 대화 로그 형식으로 시간순 출력되며, AI 응답 하단에는 평가 점수와 피드백이 시각적으로 구성되어 함께 제시된다. 학습자는 반복적으로 질의-응답-피드백 구조를 경험함으로써 자기 설명과 메타인지적 사고를 자연스럽게 유도 받고, 시스템이 제공하는 자동 피드백을 기반으로 학습 이해도를 점검하고 보완할 수 있다.

결과적으로, 본 인터페이스는 3.3절의 평가 엔진과 결합하여 학습자-GPT-4 튜터 간 상호작용의 효율성, 반복성, 피드백 활용성을 실질적으로 높이는 핵심 채널로 기능하며, 실시간 자동 평가 시스템의 교육 적용 가능성을 강화하는 기반을 제공한다.

3.5 Design of Programming Education System

지금까지 GPT-4를 기반으로 역할극 시뮬레이터의 구조(3.1절), 학습자 페르소나와 시나리오의 구성 방식(3.2절), 루브릭 기반 자동 평가 및 피드백 엔진(3.3절), 그리고 Streamlit 인터페이스를 통한 사용자 상호작용 환경(3.4

절)을 단계별로 기술하였다. 본 절에서는 이러한 구성 요소들을 통합하여 프로그래밍 교육 시스템으로 구현한 결과를 중심으로 전체 시스템의 아키텍처 설계를 설명한다.

제안된 시스템은 학습자의 자연어 질의 입력부터 GPT-4의 튜터 역할 수행, 응답 생성, 평가자 역할 전환, 루브릭 기반 평가 점수화 및 피드백 제공에 이르기까지 학습의 모든 과정을 자동으로 처리한다. 시스템 구조는 Fig. 2와 Fig. 3을 통해 시각화되며, 각각은 전체 정보 흐름 및 GPT-4의 이중 역할 수행 과정을 보여준다.

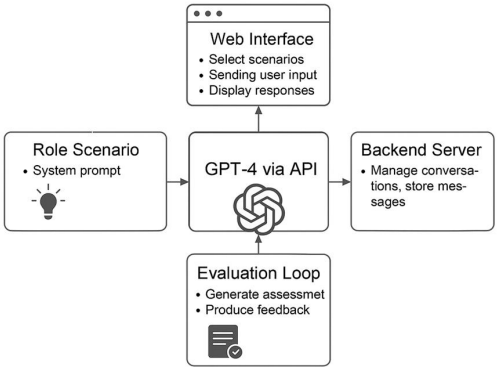

Fig. 2. GPT-4-based System Architecture

Fig. 2는 시스템의 상위 구성 요소 간 흐름을 나타내며, 학습자는 Streamlit 기반의 웹 인터페이스에서 시나리오 유형과 학습 수준을 선택하고 질문을 입력한다(Role Scenario). 해당 질의는 GPT-4 API에 전달되어 프롬프트에 따라 응답이 생성되고(Backend Server), 생성된 응답은 평가자 역할로 전환된 GPT-4에 의해 루브릭 항목별로 자동 평가된다(Evaluation Loop). 평가 결과는 점수와 자연어 피드백의 형태로 학습자에게 실시간 제공된다(Web Interface).

Fig. 3은 GPT-4가 튜터와 평가자 역할을 순차적으로 수행하는 흐름을 보여주며, Table 1과 같은 동적 루브릭 항목에 따라 자동 채점과 피드백이 생성되는 구조를 설명한다. 이러한 실시간 처리 및 반복 학습 흐름의 일관성을 유지하기 위해 Fig. 2의 벡엔드 서버가 FastAPI 기반으로 구축되었으며, 사용자 입력과 GPT-4 응답, 평가 결과, 피드백을 JSON 형식으로 구조화하여 저장하고, 세션 흐름 관리 및 동시 사용자 요청 중계 기능을 제공한다. 또한 사용자별 피드백 이력 관리를 통해 자기 주도적 학습을 지속할 수 있도록 지원한다.

결과적으로, 생성형 AI 기반 프로그래밍 교육을 위한 실시간 역할극 시뮬레이션 및 자동 평가 시스템의 전체 구조를 완성하게 된다. 이 시스템은 단일 응답 수준을 넘어서 반복 학습 기반의 상호작용과 메타인지 유도를 가능하게 하며, 실질적인 교육 현장에 적용할 수 있는 자동화된 교수-학습 플랫폼의 모델을 제시한다.

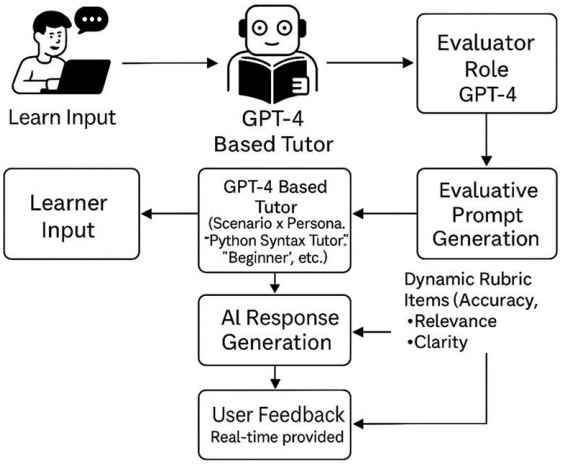

Fig. 3. GPT-4-based AI Tutor System Architecture

3.6 System Development Environment

앞서 제안한 GPT-4 기반 역할극 시뮬레이터 및 자동 평가 시스템은 클라우드 기반 환경에서 구현되었으며, 개발 플랫폼으로 Google Colab과 Streamlit Cloud를 병행 활용하였다. 전체 시스템은 Streamlit 웹 UI, FastAPI 백엔드, 그리고 GPT API 호출 로직을 중심으로 구성되었다.

GPT-4가 시뮬레이션 튜터와 자동 평가자의 이중 역할을 하기 위해, 각 역할에 맞는 프롬프트 템플릿 구조가 함께 구현되었다. 튜터 역할 프롬프트 템플릿은 학습자 수준과 시나리오 정보를 포함하여, GPT-4가 문법 설명, 예시 제공, 단계별 안내 등을 포함한 응답을 생성하도록 유도한다. 평가자 역할 프롬프트 템플릿은 사전에 정의된 루브릭 항목에 따라 튜터의 응답을 정량 평가하고, 항목별 평가를 자연어로 생성하도록 구성되었다. Fig. 4는 실제로 사용된 Python 기반 CoT 프롬프트 템플릿 예시이다.

이러한 프롬프트 엔지니어링 구조는 역할에 따른 응답의 일관성을 높이고, 자동화된 정량 평가의 타당성과 신뢰성을 향상하는 핵심 요소로 기능한다. 특히, 다양한 평가 시나리오를 처리하고 복수의 프롬프트 체인을 동적으로 구성하기 위해 LangChain 프레임워크가 백엔드 서버 시스템에 추가적으로 통합되었다. LangChain은 아래와 같은 역할로 확장성을 높였다:

• 프롬프트 템플릿 구성 자동화: 시나리오 유형, 학습자 페르소나에 따라 프롬프트 체인을 동적으로 구성할 수 있도록 LangChain의 PromptTemplate, LLMChain을 활용하였다. 이를 통해 GPT-4 기반 튜터와 평가자의 역할

```
# Tutor Prompt Template
tutor_prompt = f"""
You are a tutor playing the role of {scenario}. The learner's level is {persona}.
Please respond with concept explanations, example code, and step-by-step guidance regarding the question.
[Learner Question]: {question}
"""
# Evaluator Prompt Template
evaluator_prompt = f"""
You are a GPT-4 evaluator. Please assess the following response based on the 5 rubric criteria below:
Accuracy, Relevance, Clarity, Use of Examples, Logical Flow (Each scored from 0 to 2 points).
Provide the total score, a score for each criterion, and a brief justification for each.
[Response]: {tutor_answer}
"""
```

Fig. 4. CoT Prompt Templates for Python Tutor & Evaluator Roles

전환이 자연스럽게 이루어진다.

• 루브릭 평가 체인 관리: 각 응답에 대한 루브릭 평가를 수행하기 위해 LangChain의 SequentialChain을 이용해 AI 응답 → 평가자 역할 변환 → 루브릭 채점 → 피드백 생성을 연속적으로 자동화하였다.

• 메모리 기반 대화 흐름 관리: LangChain의 ConversationBufferMemory를 적용하여 사용자 입력과 GPT-4 응답을 일관된 대화 흐름으로 관리하였으며, 학습자의 반복 학습 시 이전 맥락이 반영된 피드백 제공이 가능해졌다.

• GPT-4 모델 지원: LangChain에서 OpenAI LLMWrapper를 통해 GPT-4와 GPT-4o, GPT-5 등을 유연하게 선택할 수 있으며, system/user 프롬프트를 조합하여 역할 분리가 명확한 교육 시나리오를 구현하였다.

또한 FastAPI 기반의 백엔드 서버는 LangChain에서 생성된 체인 실행 결과를 수신하고, 이를 JSON 로그로 저장하며, 사용자가 Streamlit 웹 인터페이스에서 입력한 학습 질문과 평가 결과를 실시간으로 중계하는 역할을 수행하였다. 서버는 uvicorn, pydantic, requests, os, datetime 등의 모듈과 함께 LangChain 체인 호출을 위한 AsyncRunnable, agent executor 등을 포함하도록 구성되었다.

이와 같은 구조는 향후 다양한 교육 모듈을 신속하게 확장할 수 있는 유연한 기반을 제공하며, 실시간 역할극 기반 학습과 자동 평가를 LangChain 체인 조합 방식으로 체계화함으로써, 교육용 LLM 시스템 설계의 새로운 방향을 제시한다.

IV. Experiments and Analyses

앞서 3장에서 제안한 생성형 AI 기반 프로그래밍 교육 시스템이 실제 프로그래밍 교육적 효과가 있는지를 검증하기 위해 수행한 실험의 설계 및 분석을 다룬다. 실험은 정량적 루브릭 평가 지표, 평가 일관성, 반복 학습에 따른 변화를 중심으로 다각적으로 진행되었다.

4.1 Experimental Data Structure Based on Prompt Engineering

본 절에서는 GPT-4 기반 역할극 시뮬레이션과 자동 루브릭 평가 시스템이 프롬프트 엔지니어링을 통해 어떻게 학습 데이터를 구성하고 실험 설계를 실행했는지를 기술한다. 제안된 시스템은 GPT-4가 튜터와 평가자 역할을 모두 수행하도록 설계되었으며, 이 과정은 프롬프트 템플릿을 중심으로 자동화되었다.

실험은 각 시나리오 유형(Python 튜터, Pandas 분석 조교, 객체지향 개념 설명자)과 학습자 페르소나(초급, 중급, 고급)를 조합하여 구성되었으며, 대표 사례로 Python 튜터 시나리오와 초급 수준 페르소나를 설정하였다. 총 10회차 세션 동안 반복적으로 문답 데이터를 생성하였고, 이에 따른 응답은 루브릭 기반 자동 평가 시스템을 통해 점수화되었다. Pandas 및 OOP 시나리오와 상위 수준 페르소나의 데이터도 동일 방식으로 수집되었으나, 지면상의 한계로 해당 사례는 생략한다.

Table 2는 수집된 데이터 중 일부로, 자연어 질의-응답 구조-피드백과 루브릭 점수화 결과를 함께 나타낸다. 학습자는 변수 선언 오류, 반복문 구조, 출력 오류 등과 관련된 질문을 입력하였고, GPT-4는 튜터 역할을 수행하며 개념 설명, 예시 제공, 단계별 안내를 포함한 응답을 생성하였

Table 2. A Sample Interaction and Rubric Feedback in Python Tutor Scenario

Session No	Learner Question	GPT-4 Tutor Response	Rubric score	Rubric Feedback
1	변수를 선언했는데 에러가 나요	변수 이름에 숫자가 먼저 오거나 공백이 들어간 경우일 수 있어요. 예: user_name = "Tom"	8/10	오류 유형을 잘 설명했으나, 오타 예시가 추가되면 더 좋음.
2	print랑 return의 차이가 뭐에요?	print는 결과를 화면에 출력하고, return은 함수 결과를 반환합니다. 예: def f(): return 3	8/10	예시는 명확하나 함수 문맥 설명이 다소 부족함.
3	리스트 정렬은 어떻게 하나요?	list.sort() 또는 sorted(list)를 사용할 수 있어요. 예: sorted([3,1,2]) → [1,2,3]	9/10	대안 제시와 예시 설명 모두 충실함.
4	for문 안에 if를 넣어도 되나요?	가능합니다. 예: for i in range(5): if i % 2 == 0: print(i)	9/10	문법 구조 설명과 예시 모두 명확함.
5	while문이랑 for문의 차이는요?	반복 횟수가 정해지면 for, 조건 반복이면 while이 적합합니다. 예: while i < 5: ...	9/10	비교 설명과 예시 모두 적절하며, 초급자의 이해 수준에 잘 맞춤.

다. 이후 동일 모델이 평가자 역할로 전환되어 응답을 자동 평가하였으며, Table 1에 정의된 다섯 가지 루브릭 항목에 따라 점수화되었다.

또한, Table 2는 학습자 페르소나의 질문이 어떻게 발전해 가는지를 보여줄 뿐만 아니라, AI 튜터 응답의 평가 품질과 피드백이 루브릭 기반 자동화 방식으로 일관되게 생성되고 있음을 시각적으로 뒷받침한다. 초기 세션에서는 “변수를 선언했는데 에러가 나요”와 같이 비정형적이고 모호한 질의가 생성되었으나, 후반 세션으로 갈수록 “for 문 안에 if를 넣어도 되나요?” 또는 “리스트를 range 함수로 초기화할 수 있나요?”와 같은 보다 명확한 문법 구조 기반의 질문이 생성되었다. 이러한 결과는 프롬프트 기반 역할 시뮬레이션을 통해 GPT-4가 단순한 응답 생성기를 넘어, 학습자 페르소나의 수준에 맞는 질의 흐름을 점진적으로 구성할 수 있음을 보여준다. 또한 전체적으로 응답 품질은 세션이 반복됨에 따라 정교화되는 경향을 보였으며, 이는 시뮬레이션 기반 반복 학습이 AI와 학습자 간 상호작용의 질을 향상시킬 수 있는 가능성을 시사한다.

4.2 Quantitative Fixed Rubric Evaluation

앞서 4.1절에서 구성된 실험 데이터를 바탕으로, 본 절에서는 고정형 루브릭 기준에 따른 정량적 평가 결과를 분석하였다. 여기서 고정형 루브릭이란 사전에 정의된 평가 항목과 점수 기준에 따라 학습자의 응답을 정량적으로 채점하는 방식으로, 평가 기준의 일관성과 객관성을 확보하는 데 유용하다.

본 연구에서는 Table 1의 5개 평가 항목—정확성(Accuracy), 관련성(Relevance), 명확성(Clarity), 예시 사용(Use of Example), 논리 흐름(Logical Flow)—으로 구성된 고정형 루브릭을 GPT-4 평가자에게 적용하였다. 실험은 Python 문법, Pandas 데이터 분석, OOP 개념 등 3가지 시나리오와 학습자 수준(초급, 중급, 고급)을 조합하여 수행되었으며, 각 조건별 10회 반복 세션, 총 50개 문답이 수집되었다. 전체 450개의 응답에 대해 GPT-4는 평가자 역할로서 Table 1에 정의된 5개 항목을 기준으로 0~2점 척도의 점수를 산정하고, 총점 10점 만점의 루브릭 기반 자동 평가를 수행하였다.

Table 3은 시나리오 유형별 평가 결과를 요약한 것으로, Python 튜터가 평균 8.8점으로 가장 높은 점수를 기록하였다. 이 시나리오는 단계적 설명과 예시 활용에 강점을 보였으나, 일부 시간 개념에서 예시가 누락되는 경향이 있었다. Pandas 데이터 분석 시나리오는 평균 8.2점으로 시각화 예시 부족이 반복적으로 지적되었고, OOP 개념 설명자는 평균 8.4점을 기록했으나 추상 용어 사용 시 구체적 사례 제시 부족이 감점 요인으로 작용하였다.

Table 3. Fixed Rubric Evaluation by Scenario Type

Scenario Type	Average Score (out of 10)	Key Issues Identified
Python Tutor	8.8/10	Missing timing check examples in some responses
Pandas Data Analysis	8.2/10	Lack of visualization examples
OOP Concept Explainer	8.4/10	Insufficient concrete examples when using abstract terminology

Table 4는 학습자 페르소나 수준별 평가 결과를 정리한 것이다. 고급 학습자는 평균 8.6점(±1.0)으로 가장 높은 성과를 보였으며, 초급은 8.4점(±1.1), 중급은 8.2점(±1.2)을 기록하였다. 표준편차는 전반적으로 ±1.0 내외로, 응답 채점의 일관성을 반영하였다. 초급 수준에서는

예시 제공 여부가 점수에 큰 영향을 미쳤고, 중급은 용어 선택과 설명 단계의 명확성이 주요 감점 요소였다. 반면 고급 학습자는 보다 복잡한 질의에 대해서도 GPT-4가 비교적 정제된 응답을 생성함으로써, 모델의 고차원 개념 대응력이 평가에 긍정적 영향을 준 것으로 해석된다.

Table 4. Fixed Rubric Evaluation by Scenario According to Persona Level

Persona Level	Scenario Type	Number of Sessions	Total Responses	Average Score (±SD)
Beginner	Python Tutor	10	50	8.4 (±1.1)
Intermediate	Pandas Data Analysis	10	50	8.2 (±1.2)
Advanced	OOP Concept Explainer	10	50	8.6 (±1.0)

이러한 결과는 본 시스템이 시나리오 및 학습자 수준에 따라 구조화된 평가 기준을 일관되게 적용할 수 있으며, 실시간 피드백과 자동 평가의 실용성을 갖추고 있음을 보여준다. 다만 시나리오별 질의 특성과 응답 맥락에 따라 항목별 점수 편차가 존재하므로, 평가 프롬프트의 맥락 기반 세분화 및 항목 가중치 조정이 보완되어야 한다.

4.3 Adaptive Rubric Evaluation Based on Scenario and Learner Level

앞선 4.2절에서는 생성형 AI 응답을 정량적으로 평가하기 위해 Table 1에서 보여준 5가지 고정형 루브릭 항목을 기반으로 점수를 산출하였음을 설명하였다. 고정형 루브릭은 평가 기준의 일관성과 비교 가능성을 보장한다는 장점이 있으나, 모든 학습 상황, 시나리오 유형, 혹은 학습자 수준에 동등하게 적용하기에는 유연성이 부족하다는 한계가 존재 한다. 예를 들어, Python 문법 설명 시나리오에서는 명확한 예시와 단계적 설명이 중요하지만, 객체지향 개념 설명에서는 개념적 깊이와 구조적 일관성이 더 중요하게 평가되어야 한다. 초급 학습자의 경우 설명의 이해 용이성과 예시 제시가 핵심이고, 고급 학습자에게는 응답의 개념 정확성과 논리 구조가 우선되어야 할 수 있다. 또한 Pandas 분석 시나리오에서는 시각화와 데이터 해석 항목이 중요하게 평가되어야 하는 등 과제 특성과 학습자 수준에 따라 요구되는 평가 기준이 달라진다.

이를 보완하기 위해 본 연구에서는 시나리오, 과목 유형, 학습자 수준에 따라 평가 항목과 가중치를 동적으로 구성할 수 있는 “적응형 루브릭 생성기(Adaptive Rubric Generator)“를 설계하였다. 이 모듈은 학습자가 선택한 시나리오 유형(예: 문법 설명, 디버깅, 추상 개념 설명), 페르소나 수준(초급, 중급, 고급)에 따라 평가 항목을 선별하고, 항목별 가중치를 자동으로 매핑한다. Fig. 5의 예시 코드에서와 같이 각 시나리오(task_type)에 따라 평가 항목이 달라지고, 학습자 수준(learner_level)에 따라 항목별 비중(weight)도 달라지도록 구현하였다.

```
# Adaptive Rubric Generator
def generate_rubric(task_type, learner_level):
    rubric_pool = {
        'Python Grammar Tutor': \
          ['Accuracy', 'Clarity', 'Use of Examples', 'Logical Flow'],
        'Pandas Assistant': \
          ['Data Relevance', 'Use of Visualization', 'Interpretation Logic'],
        'OOP Concept Explainer': \
          ['Class Design', 'Level of Abstraction', 'Maintainability']
    }

    weight_matrix = {
        'Beginner': [1.5, 1.2, 1.0, 1.5],
        'Intermediate': [1.0, 1.0, 1.0, 1.0],
        'Advanced': [1.0, 1.0, 1.2]
    }

    items = rubric_pool.get(task_type, \
          ['Accuracy', 'Clarity', 'Use of Examples', 'Logical Flow'])
    weights = weight_matrix.get(learner_level, [1.0] * len(items))
    return list(zip(items, weights))
```

Fig. 5. Scenario-Based Use of the Adaptive Rubric generator

이러한 적응형 루브릭은 실제 Streamlit 앱에 통합되어 자동 평가 루프와 연계되며, 사용자 UI에서 시나리오 및 페르소나 선택 시 자동으로 구성된다. 특히 이 구조는 향후 자연어 피드백 생성 시에도 연동되어, 각 항목에 대한 평가 근거 및 개선 제안을 출력할 수 있다.

결과적으로, 적응형 루브릭은 고정형 루브릭의 평가 일관성과 비교 가능성을 유지하면서도 다양한 교육 맥락에 따라 평가 항목을 유연하게 구성할 수 있어 평가의 타당성과 피드백의 개인화를 동시에 충족시킬 수 있다. 이는 AI 기반 자동 평가 시스템이 단순 점수화 도구를 넘어, 진단 중심의 교육 피드백 시스템으로 진화하는 데 핵심적인 기반이 될 수 있다.

4.4 Effects of Repetitive Learning with Adaptive Rubric Evaluation

본 절에서는 4.3절에서 제안한 적응형 루브릭 기반 자동 평가 시스템이 반복 학습 상황에서 학습자 페르소나별 GPT-4의 응답 품질에 어떠한 영향을 미쳤는지를 실험적으로 분석하였다. 실험은 세 수준의 학습자 페르소나(Beginner, Intermediate, Advanced)가 각기 다른 시나

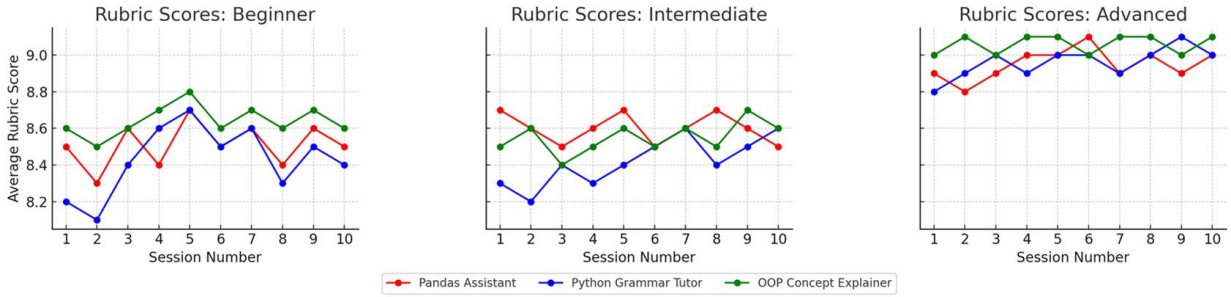

Fig. 6. Rubric Scores Trends per Session by Persona

리오(Python Tutor, Pandas Assistant, OOP Concept Explainer)와 매칭되어 총 10회 세션을 반복 수행한 결과를 바탕으로 한다.

Fig. 6은 세션 번호에 따른 평균 루브릭 점수의 추이를 시나리오별로 시각화한 것으로, 반복 학습을 통한 적응형 루브릭 점수 변화 양상을 확인할 수 있다. 우선 초급 학습자(Beginner)의 경우, 실험 초반에는 Pandas Assistant 시나리오에서 상대적으로 낮은 점수를 보였으나, 반복 세션이 진행됨에 따라 점수 분산이 줄고 모든 시나리오에서 8.2점 이상으로 수렴하는 경향이 나타났다. 이는 피드백을 통한 개념 정교화와 자기설명 전략의 향상을 보여주는 결과이다.

중급 학습자(Intermediate)는 전체적으로 8.2~8.8점 수준에서 안정된 점수 유지를 보였으나, 일부 세션에서는 시각화 관련 질의가 포함된 Pandas 시나리오에서 점수 하락이 관찰되었다. 이는 질의의 복잡성이 증가하면서 AI의 일관된 응답 품질 유지가 다소 어려워지는 양상을 반영한다.

고급 학습자(Advanced)는 전 세션에 걸쳐 8.8점 이상의 높은 점수를 유지하였으며, 특히 OOP 개념 설명 시나리오에서는 점수 변동이 거의 없이 일관된 응답 품질을 보였다. 이는 GPT-4가 고차원 추상 개념에 대해 높은 정합성과 명확성을 유지할 수 있다는 점을 시사한다.

이러한 결과는 반복 피드백 구조와 적응형 루브릭 평가 시스템이 함께 작동할 때, 학습자의 수준별 이해도와 사고 전략이 점진적으로 개선될 수 있음을 실증적으로 보여준다. 초급 학습자에게는 명확한 예시 제공과 구조화된 피드백이 개념 정착에 효과적이었으며, 중급·고급 학습자에게는 AI의 응답 정교도와 논리 흐름이 유의미한 학습 효과를 유발하였다.

결론적으로, 적응형 루브릭 평가 시스템은 학습자의 수준별 니즈에 맞는 평가 항목과 가중치를 반영함으로써 반복 상황에서도 일관성과 개인화된 피드백 제공이라는 두 가지 목표를 동시에 실현할 수 있음을 보여준다.

4.5 Reliability and Validity of the Automated Evaluation System

본 절에서는 제안한 자동 평가 시스템의 일관성과 신뢰도를 정량적으로 검증하고, 인간 평가자와의 비교 분석을

Table 5. An Example of Rubric Evaluation Criteria for Python Tutor

Evaluation Item	Evaluation Criteria	Score 0	Score 1	Score 2
Accuracy	Does the explanation provide accurate information relevant to the learner's question?	Unclear explanation or includes obvious errors	Generally correct but contains minor errors	Concepts and code are accurately explained
Relevance	Is the answer appropriate to the subject and context of the question?	Includes irrelevant or off-topic information	Related to the question but includes vague or partial content	Clear explanation aligned with context and question intent
Clarity	Is the explanation structured in a step-by-step and easy-to-understand manner?	Too abstract or incoherent sentence structure	Flow is maintained but some parts lack clarity or cohesion	Clear and structured step-by-step explanation
Use of Example	Are real examples or code effectively included?	No example or irrelevant example	Example lacks detail or clarity	Includes specific code or examples of key concepts
Logical Flow	Is the explanation logically structured and coherently connected?	Disconnected or fragmented logic	Has logical flow but weak connections between sentences	Consistent and well-structured logical flow

통해 평가 타당성을 실증하였다. 특히, 자동 평가가 반복 상황에서도 안정적인 결과를 제공할 수 있는지를 확인하고자 하였다. 여기서는 Python 시나리오와 초급 학습자 페르소나를 중심으로 사례를 제시하였으며, Pandas 데이터 분석 조교 및 OOP 개념 설명자 시나리오에 대한 세부 학습자 페르소나 분석은 지면 관계상 생략하였다.

시스템 내부의 자동 평가 모듈은 GPT-4를 평가자 역할로 활용하며, Python 튜터 시나리오에서 생성된 응답을 대상으로 Table 5의 루브릭 항목에 따라 0~2점으로 점수화한다. 총점은 10점 만점이며, 각 항목은 교육적 설명력과 문법적 구조, 예시 활용 등 실제 학습자 이해도와 직결된 기준으로 구성되었다.

시스템의 평가 일관성을 검증하기 위해 동일한 AI 응답에 대해 자동 평가를 두 차례 반복 수행한 결과, 항목별 평균 점수 차이는 ±0.5점 이내로 나타났다. 이는 프롬프트 기반 자동 평가 구조가 반복 평가 상황에서도 일관된 기준을 유지함을 의미한다.

평가 타당성 검증을 위해 인간 평가자와의 비교 실험도 수행되었다. Python 프로그래밍 교육을 수강한 대학생 평가자 2인이 총 15개의 AI 응답을 동일한 루브릭 기준으로 독립 채점하였으며, 항목별 평가 근거도 함께 서술하도록 하였다.

그 결과, Fig. 7을 보면 평가자 간 평균 일치율은 92%였으며, 인간 평가자와 자동 평가 시스템 간 점수 유사도는 91.3%로 나타났다. 특히, Accuracy와 Relevance 항목에서는 거의 동일한 평가 경향이 관찰되었고, Clarity 및 Use of Example 항목에서도 ±0.15점 이내의 편차를 보였다. 다만, Logical Flow 항목에서는 문장 전개의 논리적 일관성에 대한 인간 평가자의 보수적인 판단이 상대적으로 두드러졌다.

이러한 결과는 GPT-4 기반 자동 평가 시스템이 정해진 루브릭 항목을 정량적으로 해석하고 점수화하는 데 있어 높은 일관성과 신뢰도를 확보하고 있음을 실증적으로 보여준다.

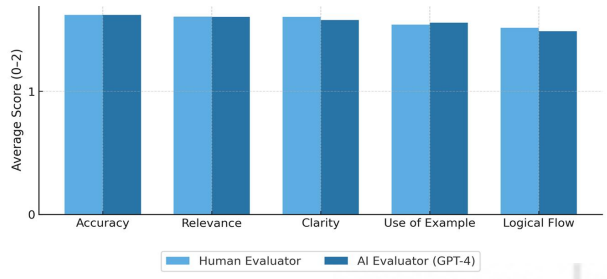

Fig. 7. Comparison of Rubric Scores Human vs AI Evaluator

V. Conclusions

본 연구는 생성형 AI 기반의 역할극 시뮬레이션과 적응형 루브릭 기반 자동 평가 모듈을 통합한 맞춤형 프로그래밍 교육 시스템을 제안하였다. 본 시스템은 GPT-4가 대화 튜터와 평가자의 이중 역할을 하도록 프롬프트 엔지니어링을 활용하여, 프로그래밍 교육 시스템을 설계·구현하고, 그 효과성과 실현 가능성을 실험적으로 검증하였다.

시스템은 학습자의 자연어 질의에 대해 GPT-4가 튜터 프롬프트 기반 응답을 제공하고, 평가 프롬프트로 전환되어 루브릭 기반 평가 점수와 피드백을 자동 산출하는 구조를 통해 단순 채점형 응답을 넘어, 생각의 연결고리(CoT, Chain of Thought)을 유도하는 자기설명 기반 학습 구조를 구현하였다. 또한, 시스템의 핵심 기여 중 하나는 적응형 루브릭 생성기 모듈의 도입이다. 이 모듈은 시나리오 유형과 학습자 페르소나 수준에 따라 평가 항목과 가중치를 동적으로 생성함으로써, 고정형 루브릭의 일률성을 보완하고 교육 맥락에 맞는 유연한 자동 평가를 가능하게 했다. 이를 통해 정량적 평가의 타당성과 실시간 피드백의 적합성을 동시에 확보할 수 있었다.

실험 결과는 이 시스템이 반복 학습 상황에서 학습자의 질의 구조와 AI 응답의 완성도를 동시에 향상시키는 효과를 입증하였다. 자동 평가 결과는 사람 평가자와 약 91%의 일치율을 보이며, 명확성 및 예시 사용 항목에서 특히 높은 평가 일관성을 보였다. 세션별 평균 점수 추이는 초급 학습자의 질의가 점차 구조화되고 고급 학습자의 응답 평가가 안정적으로 유지되는 패턴을 보여, 반복 학습 기반 자기주도 학습 효과가 실증되었다.

다만, 본 연구는 가상의 학습자 페르소나를 중심으로 시뮬레이션 실험을 진행하였기에, 실제 학습자를 대상으로 한 장기적 검증이 필요하며, 코드 실행 결과 분석이나 시각화 피드백 등 멀티모달 기능의 통합은 향후 과제로 남는다.

결론적으로 본 시스템은 생성형 AI의 교육적 활용 가능성을 실증적으로 입증하였으며, 역할 기반 상호작용, 적응형 평가, 자기설명 중심 피드백 제공이라는 세 가지 교육적 요소를 효과적으로 통합한 시뮬레이션 기반 프로그래밍 학습의 새로운 모델을 제안하였다. 이는 향후 다양한 도메인에서 AI 기반 교육 시스템 개발과 확산에 중요한 기반이 될 수 있다.

ACKNOWLEDGEMENT

This work was supported by the Shihan University Research Fund, 2025.

REFERENCES

[1] Codecademy, "Learn to code interactively, for free," 2024. https://www.codecademy.com

[2] LeetCode, "Online Judge & Coding Platform," 2024. https://leetcode.com

[3] Kaggle, "Datasets and Competitions for Machine Learning," 2024. https://www.kaggle.com

[4] Z. Chu, S. Wang, J. Xie, T. Zhu, Y. Yan, J. Ye, A. Zhong, X. Hu, J. Liang, P. Yu, Q. Wen, "LLM Agents for Education: Advances and Applications," March 2025, https://doi.org/10.48550/arXiv.2503.11733

[5] E. Mollick, L. Mollick, Y. Hu, "AI Agents and Education: Simulated Practice at Scale," July 2024. https://doi.org/10.48550/arXiv.2407.12796

[6] H. Ahn , "Exploring the Educational Applications of AI-Based Feedback: Focusing on the Beginner Learner Corpus," *Bilingual Research,* Vol. 10, No. 96, pp. 63-95, June 2024, DOI: 10.17296/korbil.2024..96.63.

[7] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. Le, "Chain-of-Thought Prompting Elicits Reasoning in Large Lnguage Models," Jan 2023, https://doi.org/10.48550/arXiv.2201.11903

[8] Z. Zhan, X. Jia, L. Lin, Q. He, Y. Wang, "CoderAgent: Simulating Student Behavior for Personalized Programming Learning with Large Language Models," May 2025, *Proceedings of the 2025 ACM Conference on Learning*, https://doi.org/10.48550/arXiv.2505.20642

[9] J. Baktash, M. Dawodi, "Gpt-4: A Review on Advancements and Opportunities in Natural Language Processing," May 2023. https://doi.org/10.48550/arXiv.2305.03195

[10] E. Ross, Y. Kansal, J. Renzella, A. Vassar, A. Taylor, "Supervised Fine-Tuning LLMs to Behave as Pedagogical Agents in Programming Education," Feb 2025. https://doi.org/10.48550/arXiv.2502.20527

[11] T. Wang, N. Zhou, Z. Chen, "Enhancing Computer Programming Education with LLMs: A Study on Effective Prompt Engineering for Python Code Generation," July 2024, https://arxiv.org/pdf/2407.05437

[12] B. Cowan, Y. Watanobe, A. Shirafuji, "Enhancing Programming Learning with LLMs: Prompt Engineering and Flipped Interaction," March 2024, https://doi.org/10.1145/3634814.3634816

[13] T. Phung, N. Kotalwar, M. Liut, J. Leinonen, P. Denny, "Humanizing Automated Programming Feedback: Fine-Tuning Generative Models with Student-Written Feedback," Sep 2025, https://doi.org/10.48550/arXiv.2509.10647

[14] E. Kasneci, B. Becker, K. Sessler, R. Kübler, U. Schmid, "ChatGPT for Good? On Opportunities and Challenges of Large Language Models for Education," April 2023, https://doi.org/10.1016/j.lindif.2023.102274

[15] M. Park, S. Kim, S. Lee, S. Kwon, K. Kim, "Empowering Personalized Learning through a Conversation-based Tutoring System with Student Modeling," March 2024, https://doi.org/10.1145/3613905.3651122

Authors

Euhee Kim received the M.S. degree in Computer Engineering from Dongguk University, Korea, in 2002 and Ph.D. degree in Mathematics from The University of Connecticut, U.S.A in 1995.

Euhee Kim is currently a Professor in the Department of Software Convergence at Shinhan University. She is interested in AI, NLP and Big Data Computing.