JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 29 No. 8, pp. 191-200, August 2024
https://doi.org/10.9708/jksci.2024.29.08.191

A Study on Generative AI-Based Feedback Techniques for Tutoring Beginners' Error Codes on Online Judge Platforms

Juyeon Lee*, Seung-Hyun Kim**

*Student, Dept. of Computer Education, Korea National University of Education, Cheongju, Korea
**Assistant Professor, Dept. of Computer Education, Korea National University of Education, Cheongju, Korea

[Abstract]

The rapid advancement of computer technology and artificial intelligence has significantly impacted software education in Korea. Consequently, the 2022 revised curriculum demands personalized education. However, implementing personalized education in schools is challenging. This study aims to facilitate personalized education by utilizing incorrect codes and error information submitted by beginners to construct prompts. And the difference in the frequency of correct feedback generated by the generative AI model and the prompts was examined. The results indicated that providing appropriate error information in the prompts yields better performance than relying solely on the excellence of the generative AI model itself. Through this research, we hope to establish a foundation for the realization of personalized education in programming education in Korea.

▸ **Key words**: Online Judge, Intelligent Tutoring System, Personalized Education, GPT-3.5, GPT-4.0

[요 약]

컴퓨터 기술과 인공지능의 비약적인 발전이 국내 소프트웨어 교육에서도 많은 영향을 끼치고 있다. 이에 따라 2022 개정 교육과정에서도 맞춤형 교육을 요구하게 되었지만, 학교에서 맞춤형 교육을 실현하기에는 어려움이 있다. 이에 본 연구에서는 맞춤형 교육 실현을 위해 초보 학습자가 제출한 오답 코드와 오답 정보들을 활용하여 적절한 피드백 생성을 위한 프롬프트를 구성하였다. 그리고 생성형 인공지능 모델과 프롬프트 조합에 따른 정상 피드백 생성 빈도의 차이를 실제 데이터를 활용하여 분석하였다. 그 결과, 생성형 인공지능 모델 자체의 우수성보다 오답 정보를 포함한 프롬프트가 더 우수한 피드백 생성 성능을 나타내는 것을 확인하였다. 본 연구를 통해 국내 프로그래밍 교육에서 맞춤형 교육의 실현을 위한 토대가 되기를 기대한다.

▸ **주제어**: 온라인 저지, 지능형 튜터링 시스템, 맞춤형 교육, GPT-3.5, GPT-4.0

• First Author: Juyeon Lee, Corresponding Author: Seung-Hyun Kim
*Juyeon Lee (juyeon5741@knue.ac.kr), Dept. of Computer Education, Korea National University of Education
**Seung-Hyun Kim (kimsh@knue.ac.kr), Dept. of Computer Education, Korea National University of Education
• Received: 2024. 06. 03, Revised: 2024. 07. 26, Accepted: 2024. 07. 26.

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

I. Introduction

최근 컴퓨터 기술과 인공지능의 비약적인 발전으로 소프트웨어 교육의 중요성이 대두되고 있다. 해외에서는 초등학교부터 소프트웨어 교육을 이수할 수 있도록 교과를 운영하고 있으며[1], 소프트웨어 단독 교과 및 소프트웨어와 타 교과와의 융합 교육을 실시하고 있다[1]. 국내에서도 2022 개정 교육과정의 정보 교과 수업 시수를 2015 대비 2배 확충하였다[2]. 또한 교육부는 디지털 대전환 시대 교육의 비전을 「모든 교사들이 교육기술을 활용하여 '모두를 위한 맞춤 교육' 실현」으로 제시하고[3] 학교 내 스마트기기 보급, 디지털 교과서 등 맞춤형 교육에 필요한 인프라를 구축하고 있다[4].

기술의 발전을 통해 실제 교실에서도 인공지능을 활용한 맞춤형 교육이 현실화되고 있다. 맞춤형 교육에 대한 효과성은 1980년대부터 입증되었는데, Bloom의 연구에 따르면 개별학습을 받은 학생들의 성취도가 전통적인 수업을 받은 학생 성취도의 2배로 측정되었다[5]. 하지만 일반적인 공교육 현장에서 개별 학습을 실시하기에는 인적 자본의 부족으로 실현되기 어렵다. 한국교육개발원에서 실시한 교육기본통계 결과에 따르면 2023년의 학급당 학생 수는 중학교 24.61명 고등학교 22.90명이다[6]. 따라서 한 명의 교사는 수업 시간 동안 약 23명을 상대로 수업을 실시하게 되므로 일대일 맞춤형 수업은 실현될 수 없다. 실질적으로 일대일 개별학습이 이루어지기는 어려운 점을 지적하며, 보다 더 실용적이면서 비슷한 효과를 내는 방법을 찾아내는 문제가 Bloom의 2 sigma problem이다[5]. 이 문제를 해결하는 방법 중 하나로 제안된 것은 지능형 교수 시스템(Intelligent Tutoring System)이다. 이러한 시스템을 인공지능 챗봇으로 구현하여 프로그래밍 교육에 적용했을 때, 학습자의 학습 속도와 학습 능률의 향상으로 이어질 수 있다[7]. 뿐만 아니라 인공지능을 활용한 맞춤형 교육을 실시 하였을 때, 학업 성취 향상에 도움이 되며[8], 교사에게도 업무 경감, 원격 수업에서 학생의 학습행동에 대한 이해에 도움을 주는 등 긍정적인 영향을 미칠 수 있다[9].

특히, 최근의 인공지능 기술인 생성형 언어모델을 맞춤형 교육에 활용되는 연구가 다수 발표되고 있다. 생성형 언어모델은 초거대 데이터를 학습하여 학습된 내용을 기반으로 텍스트를 생성하는 인공지능의 한 종류이다. 분야를 가리지 않고 수집된 초거대 데이터를 학습하기 때문에 여러 교과에서 활용될 수 있는 여지가 크다. 생성형 인공지능의 교과 활용에 관한 연구 동향을 살펴보면 2023년 9월 기준으로 약 68건의 연구가 진행되었으나, 컴퓨터 과학 교육을 위해 생성형 인공지능을 활용한 사례는 5건에 불과하다[10]. 하지만 프로그래밍 교육에 필요한 시스템인 온라인 저지에서 생성형 언어모델을 사용한 연구는 없는 실정이다.

따라서 본 연구에서는 프로그래밍 과목에서 생성형 언어모델을 활용한 맞춤형 수업 방안을 제시한다. 구체적으로는 맞춤형 학습을 위해 생성형 인공지능이 학습자에게 제공해야 하는 피드백의 범주를 정립하고, 생성형 인공지능의 종류와 프롬프트의 변화에 따른 피드백 차이를 파악하고자 한다. 실제 프로그래밍 수업에서 필요한 피드백 데이터를 수집하기 위해 K대학의 프로그래밍 수업에서 온라인 저지 시스템을 구축하여 학습자의 프로그래밍 코드를 수집하였으며, 오류가 존재하는 코드에 대해 생성형 인공지능으로부터 피드백을 제시받았다. 그리고 생성형 인공지능이 생성한 피드백이 학습자의 오류 코드를 개선하는데 도움이 되는지를 분석하였다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 LLM이 맞춤형 교육에 적용된 사례를 살펴보고 3장에서 온라인 저지 플랫폼을 구축하여 실제 초보 학습자를 대상으로 오류 코드를 수집하고, 다양한 LLM 모델과 프롬프트로 오류 코드에 대한 피드백을 생성하여 그 차이를 측정한다. 4장에서는 연구 방법의 결과를 분석하고, 5장에서 연구 결과를 토대로 시사점과 제한점을 제시한다.

II. Preliminaries

1. Online judge

온라인 저지(Online Judge)란 동일 제약조건 하에서 인간의 개입 없이 자동으로 채점하는 시스템을 의미한다[11]. 온라인 저지는 학습자가 제출한 프로그램을 input 값에 대한 output값을 확인하여 채점하는 방식으로 동작한다. 온라인 저지 설계를 위해 필요한 모듈은 Fig 1과 같다[12]. 세부적으로는 client 모듈은 사용자의 코드를 입력받고 채점 결과를 보여주며, control 모듈은 client 모듈에서 채점을 요청하면 Judge 모듈을 호출하며 그 결과를 데이터베이스에 업데이트 한다[12]. 이러한 방식으로 서비스되는 온라인 저지 시스템은 국내에는 백준[13], 정올[14] 등이 있다.

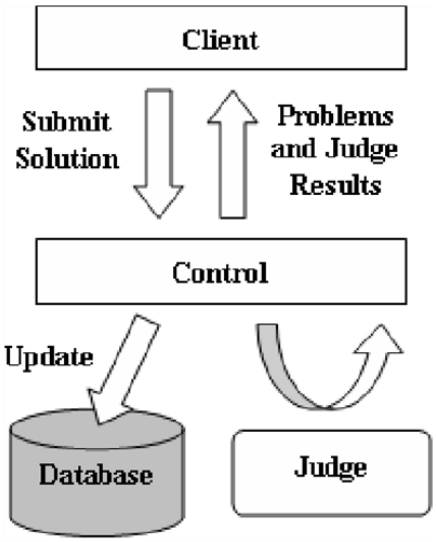

Fig. 1. Architecture of online judge[12]

온라인 저지 시스템을 활용한 프로그래밍 교육이 기존의 전통적인 프로그래밍 교육에 비해 효과적인 것은 여러 연구에서 증명되었으나, 아직 이 분야에 대한 국내 연구는 활성화 되지 않았다. 한국학술지인용색인(Korea Citation Index)에서 프로그래밍 교육을 위한 온라인 저지 연구 현황을 살펴본 결과, 총 18건에 불과한 수치를 나타내고 있다. 특히 시스템 설계/개발(4건)[15-18] 또는 사용되는 문제(문제은행 개발, 문제 분석, 문제 추천)에 관한 연구(4건)[19-22]가 다수 였다. 본 연구에서 다루고자 하는 자동 튜터링을 다룬 연구는 1건[23]에 불과하였다.

Table 1. Domestic research trends related with online judge

Category	Count
System Design/Development	4
Utilization	3
Effectiveness	3
Problem Bank	2
Testcase	1
Prediction of submission results	1
Analysis of problem	1
Problem recommendation	1
Evaluation Methods	1
Tutoring	1

문현수 외[23]의 연구에서는 제출한 프로그램에 대하여 튜터링 방법을 Compile Error, Runtime Error 등 온라인 저지에서 발생할 수 있는 여러 오류 유형에 대한 튜터링 방법을 힌트의 개념으로 제안하였다. Compile Error, Runtime Error의 경우 규칙에 따라 정형화된 코딩 가이드를 제시하였고 Wrong Answer(코드 자체의 결함은 없으나, 도출된 출력 값이 정답이 아닌 경우)결과에는 오답 소스코드와 가장 유사한 정답 소스코드에서 사용된 자료형 사용량을 비교하여 적절한 자료형을 추천하는 방법으로 코딩 가이드를 제시하였다[23]. 하지만, 위 연구는 정형화된 코딩가이드와 자료형을 추천한다는 점에서 초보 학습자에게 나타나는 예외적인 오류 유형에 대응하기 어려우며, 학습자의 이해 수준에 맞추어 튜터링 하지 못한다는 한계가 존재한다.

2. Intelligent programming tutor

IPT(Intelligent Programming Tutor)는 컴퓨터 프로그래밍을 가르치는 목적으로 개발된 지능형 교수학습 시스템을 의미한다[24]. 맞춤형 교수 학습의 효과성이 Bloom의 연구[5]를 통해 입증되면서 프로그래밍 교육에서도 맞춤형 교수 학습이 가능하도록 지능형 교수 시스템 개발에 관한 연구가 진행되어왔다. IPT의 일반적인 특성으로 '프로그램에 대한 피드백'이 있다[24]. 이러한 피드백은 인간 교사처럼 초보 학습자를 대상으로 피드백의 제공하면서 IDE/콘솔에서 발생하는 에러메시지 이상의 피드백을 제공해야 한다[24].

IPT의 초기 연구는 인공지능 프로그래밍 언어인 LISP를 효과적으로 가르치기 위해 실시되었으며, 학습자가 프로그램 작성과정에서 발생하는 문제들에 피드백을 제공하도록 개발되었다[25]. 그 결과 인간 튜터가 컴퓨터 튜터보다는 효과적이지만, 컴퓨터 튜터는 앞으로 더욱 발전할 수 있는 여지가 크고 효과 측면에서 큰 차이를 보이지 않는다는 점에서 의의를 발견하였다[25]. 이를 증명하듯 최근 연구에서는 컴퓨터 튜터의 능력이 인간 튜터 만큼 성능이 향상되었다.

김송희와 장윤제(2023)는 AI 챗봇(AI 헬피)을 활용하여 프로그래밍 교육을 실시하였고 그 결과 프로그래밍 학습 시에 오류를 수정하는 시간이 줄면서 학습 속도와 능률이 향상되는 긍정적인 효과를 확인하였지만[7], 챗봇의 경우 학습자의 질문 수준에 따라 결과가 상이하게 달라질 수 있다. 최승윤 외(2023)의 연구에서는 코드 리뷰 학습 환경을 GPT 프롬프트를 사용하는 챗봇 형태로 구축하였으며 개발된 챗봇이 맞춤형 프로그래밍 학습을 위한 도구로 사용될 가능성을 제시하였지만[26], 실제 학습데이터로 검증이 되지 못했다는 한계가 있다. 이에 본 연구는 정형화된 프롬프트를 사용하여 실제 학습과정에서 발생한 오류 코드를 대상으로 생성형 인공지능을 사용한다는 점에서 차이가 있다.

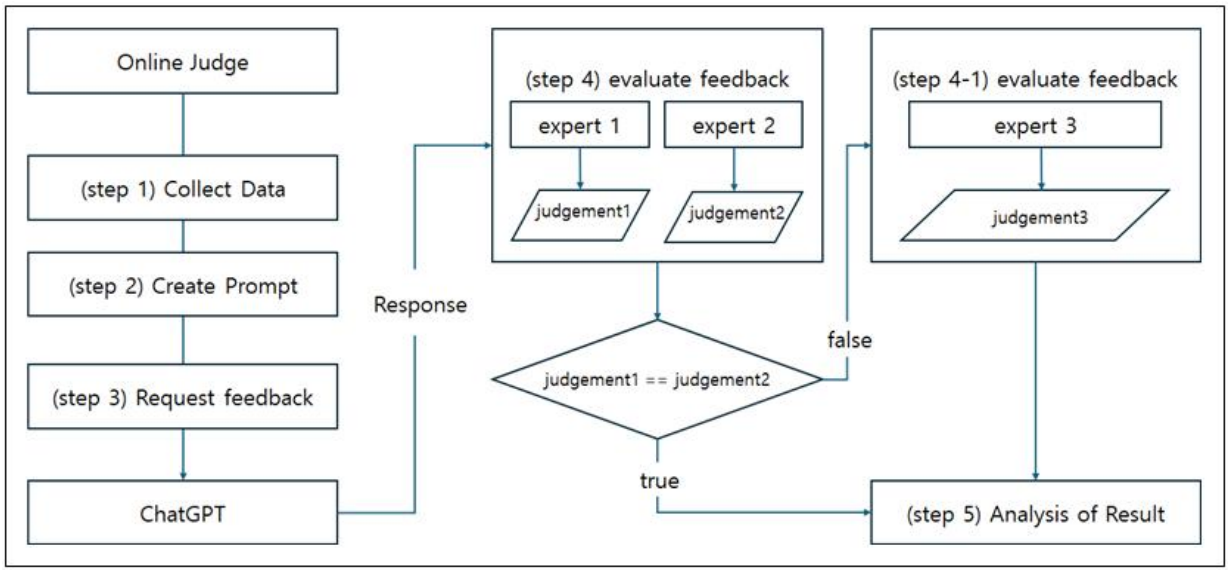

Fig. 2. Research procedure

III. The Proposed Scheme

1. Hypothesis

생성형 인공지능이 프로그래밍 튜터로서 기능하기 위해서는 프로그램 또는 코드가 사전에 학습이 되어야 한다. 그리고 사전에 학습이 되어있는 상태의 인공지능 모델에 적절한 프롬프트를 제공하여야 한다. 이때 프롬프트를 구성하는 정보에 따라 다양한 결과를 도출한 연구들이 다수 있다[27][28]. 또한, 코드 생성에 있어서 LLM의 기능을 평가하는 HumanEval Benchmark 점수를 확인하였을 때, 다양한 LLM이 사용되었으며 Benchmark 점수가 다양하게 분포되어있다[29][30]. 이러한 LLM의 특성을 고려하여, 학습자에게 효과적인 피드백을 제공하는 온라인저지 시스템 구축을 위해, 적절한 생성형 인공지능 모델과 프롬프트를 찾기 위한 다음과 같은 가설을 설정하였다.

가설 1. 생성형 인공지능의 모델에 따라 초보 학습자의 프로그래밍 오류 해결을 튜터링하기 위해 생성된 피드백의 적합도에 차이가 있을 것이다.

가설 2. 프롬프트의 차이에 따라 초보 학습자의 프로그래밍 오류 해결을 튜터링하기 위해 생성된 피드백의 적합도에 차이가 있을 것이다.

2. Research procedure

연구 절차는 Fig 2과 같다. 먼저 실제 학습자를 대상으로 진행된 강의에서 데이터를 수집한다(step 1). 프롬프트 간 피드백의 차이를 확인하기 위해 오답 정보를 제공한 프롬프트와 오답 코드만을 제공한 프롬프트로 구성한다(step 2). 그 다음 구성한 프롬프트 내용에 맞춰 수집한 오류 코드와 오답 정보를 연동된 API를 통해 보내고 피드백을 요청한다(step 3). 그 다음, 생성된 피드백이 학습자가 제출한 프로그램의 오류를 개선하는 데 도움이 될 수 있는지에 대한 여부를 2명의 전문가가 판단(step 4)하고, 전문가의 의견이 일치하지 않는 경우 본 연구자가 최종 결정(step 4-1)한 뒤 전체 결과를 분석하였다(step 5).

최적의 프롬프트와 최적의 모델을 확인하기 위해 총 4번의 실험을 실시하였다. 실험 조건은 Table 2와 같다.

Table 2. Research condition

	model	error information
1	GPT-3.5-turbo	x
2	GPT-4.0-turbo	x
3	GPT-3.5-turbo	o
4	GPT-4.0-turbo	o

3. Dataset

프로그래밍 학습에 어려움을 겪는 초보 학습자가 생성한 오류 코드 데이터 셋을 구축하기 위해 오픈소스인 Qingdao OnlineJudge를 활용하여 온라인저지 환경을 구축하였다[31]. 온라인저지에 간단 사칙연산과 파이썬의 문법, Linked List 구현, Stack과 Queue, Tree 문제를 상/중/하 난이도로 문제를 제시하였다.

관리자 계정, 정답코드, 테스트 코드 등을 제외하고 총 4,885개 데이터를 수집하였다. 10% 무작위 추출하고, 에러 메시지 미생성인 경우와 에러 중에서 Time Limit Exceed를 제외하고 총 435개를 최종 데이터 셋으로 선정

하였다. 최종 선정한 데이터 셋에서는 CE(Compile Error), RE(Runtime Error), WA(Wrong Answer) 종류의 오류가 발생하였다.

전체 데이터셋(4,885건)의 분포를 확인한 결과는 Table 3과 같으며, 최종 선정한 데이터 셋의 분포는 Table 4와 같다. 전체 코드 제출 분포(Table 3)와 무작위 추출 분포(Table 4)를 비교하였을 때 비슷한 비율을 유지하는 것을 확인할 수 있어 적절히 추출된 것으로 파악할 수 있다.

Table 3. Distribution of code submissions by category and error type(total)

category	CE	RE	WA	total
syntax	1,042	906	1,462	3,410
Linked List	517	280	368	1,165
Stack, Queue	129	66	95	290
Tree	8	5	7	20
total	1,696	1,257	1,932	4,885

Table 4. Distribution of code submissions by category and error type(random sampling)

category	CE	RE	WA	total
syntax	104	79	126	309
Linked List	39	16	41	96
Stack, Queue	12	3	13	28
Tree	1	0	1	2
total	156	98	181	435

4. Prompt

2024년 5월 16일 기준, Chen et. al의 연구에서 개발된 LLM의 코드 생성에 대한 HumanEval Benchmark 점수를 확인하였을 때, 상위 10개의 LLM 모델 중 9개가 GPT-3.5 또는 4.0을 기반으로 개발되었다[29][30]. 이는 GPT모델이 코드 관련하여 사전 훈련이 적절히 되었음을 의미한다. 따라서 본 연구에서도 클라이언트가 오답을 제출한 경우 온라인저지 서버가 피드백을 요청하기 위해 OpenAI의 gpt-turbo-3.5, gpt-turbo-4.0을 사용하였다.

프롬프트는 3가지 경우(CE, RE, WA)로 나누어 구성하였다. CE와 RE와 달리 WA의 경우 에러메시지가 생성되지 않아 프롬프트를 구성할 때 오답 정보에 차이가 발생하기 때문에 나누어 프롬프트를 작성하였고, CE와 RE는 에러가 발생하는 시점이 Compile 시점의 이전 이후로 나뉘어지기 때문에 프롬프트를 나누어 작성하였다. 프롬프트에 사용되는 오답 정보는 Table 5와 같다. 문제를 구성하는 기본 정보[11]인 입출력 구성에 관한 정보를 제공하였으며, 문제를 구성하는 기본 정보는 아니지만 학습자들의 오답 코드가 정답 코드가 될 수 있는 방향을 제시하기 위해 모범답안과 오답코드, 에러메시지, 에러 유형, input 함수 사용시 주의점 등을 포함하여 프롬프트를 구성하였다. 이후 오답 코드에서 발생한 오류를 해결하기 위해 구성한 프롬프트를 OpenAI에서 제공하는 API를 사용하여 전송하였고 힌트와 에러키워드를 요청하였다. 프롬프트의 사용 예시는 Table 6와 같다.

Table 5. Error information

category	description
server code	example code
submit code	error code
error type	WA, CE, RE
error message	error message
input value	-
output value	output value of example code and error code
caution information	-

Table 6. Prompt example(include error info)

role	content
system	당신의 역할은 온라인 저지의 심판입니다.
assistant	정답 코드: def add(a,b): return a+ b a ,b = input().split() a = int(a) b = int(b) print(add(a,b))
assistant	오답 코드: A, B =map(int, input().split()) print(A-B)
assistant	발생한 오류: Wrong Answer
assistant	입력 값: 1000 2000
assistant	입력에 대한 정답코드의 출력 값: 3000
assistant	입력에 대한 오답코드의 출력 값:-1000
assistant	input 괄호 안에 문자열 넣으면 정답코드와는 다른 값이 출력됩니다.
assistant	error_keyword(정답코드의 출력 값과 오답코드의 출력 값이 다른 이유)와 hint(정답코드의 출력 값과 오답코드의 출력 값이 같아지도록 최대한 간접적으로 힌트를 주세요, 정답코드는 절대 알려주지 마세요) 키 두개를 갖는 json 형식으로 알려주세요.

프롬프트 구성에 사용한 오답 정보가 적절한지 비교하기 위해 오답 코드만으로 구성한 프롬프트도 구성하였으며 구성한 프롬프트에 사용된 기본 정보는 Table 7과 같으며, 프롬프트 사용 예시는 Table 8과 같다.

Table 7. Basic information

category	description
submit code	error code

Table 8. Prompt example(exclude error info)

role	content
system	당신의 역할은 온라인 저지의 심판입니다.
assistant	오답 코드: A, B =map(int, input().split()) print(A-B)
assistant	error_keyword(정답코드의 출력 값과 오답코드의 출력 값이 다른 이유)와 hint(정답코드의 출력 값과 오답코드의 출력 값이 같아지도록 최대한 간접적으로 힌트를 주세요, 정답코드는 절대 알려주지 마세요) 키 두개를 갖는 json 형식으로 알려주세요.

5. Analysis

서울 소재의 에듀테크 기업과, 청주 소재의 K 대학교에서 프로그래밍 캠프 강사로 활동한 경력이 있는 파이썬 프로그래밍에 능숙한 2명(전문가1, 전문가2)을 선정하여 별도의 공간에서 LLM의 피드백 결과를 분류하였다. 반환된 피드백이 오답 코드의 결함을 해결할 수 있도록 도움을 주는 경우 'O', 아닌 경우 'X'로 표기하였다. 전문가1과 전문가2의 O/X 의견이 일치하지 않을 경우 본 논문의 저자가 최종 결정을 하였다.

Table 9는 정상 피드백으로 판단된 예시를 보인다. Table 9의 submit code는 실제로 제출된 코드이며, feedback은 Table 5와 같은 오답 정보로 프롬프트를 구성하여 피드백을 요청한 결과이다. submit code에서 발생한 오류는 Runtime Error이며 입력받은 "1000 2000"의 문자열을 띄어쓰기로 분리하지 않고 바로 자료형 변환을 시도하여 발생하였다. 이러한 오류가 발생하였을 때, 생성형 인공지능은 입력받는 양식을 확인하고 문자열 분할 방법을 추천하고 있다. 따라서 정확히 코드의 문제를 파악하고 적절한 피드백을 생성한 것으로 판단하였다.

Table 10는 비정상 피드백으로 판단된 예시를 보인다. Table 10의 submit code는 실제로 제출된 코드이며, feedback은 Table 5와 같은 정보로 프롬프트를 구성하여 피드백을 요청한 결과이다. submit code에서 발생한 오류는 Wrong Answer이고 입력 값이 10을 넘을 경우 "error"라는 문자열을 출력하게 되면서 온라인 저지가 저장하고 있는 출력값과 달라지는 문제로 발생한 오류이다. 이러한 오류가 발생하였을 때, 생성형 인공지능은 입력 값을 숫자로 바꿀 것을 추천하고 있다. 이는 학습자가 온라인 저지의 문제를 해결하는 데에 도움을 주지 못하고 있으므로 비정상 피드백으로 판정하였다.

Table 9. Example of normal feedback

role	content
submit code	A = int(input("첫 번째 정수 A를 입력하세요: ")) B = int(input("두 번째 정수 B를 입력하세요: ")) result = A + B print("두 정수의 합은:", result)
feedback	입력받는 방식을 확인하세요. 띄어쓰기로 구분된 여러 값을 한 번에 처리할 수 있도록 입력 함수와 문자열 분할 방법을 조정해야 합니다.

Table 10. Example of abnormal feedback

role	content

submit code	a, b =map(int, input().split())
	if b < 10:
	  print(a + b)
	else:
	  print("error")

feedback	input() 함수는 사용자로부터 문자열을 입력받아서 반환합니다. 따라서, 입력 값을 숫자로 변환하여 계산해야 합니다. 예를 들어, int(a)와 int(b)를 통해 입력 값을 정수로 변환할 수 있습니다.

IV. Results

1. Descriptive statistics results for normal feedback about generative AI and prompt

4가지 실험결과에 따른 정상피드백 생성 빈도는 Fig 3와 같다. CE, RE, WA에서 모두 GPT-4.0 모델에 오답 정보를 제공하였을 때, 가장 우수한 성능을 나타내었다. 특히, WA의 경우 GPT-3.5 모델에 오답 정보를 제공하지 않은 경우와 비교하였을 때 약 7배 차이가 나타남을 확인하였다. 또한 전반적으로 비교하였을 때에도 최악의 경우 보다 약 3배 우수하며, GPT-3.5(오답 정보 불포함), GPT-4.0(오답 정보 불포함), GPT-3.5(오답 정보 포함), GPT-4.0(오답 정보 포함) 순으로 정상 피드백을 생성하는 비율이 높았다.

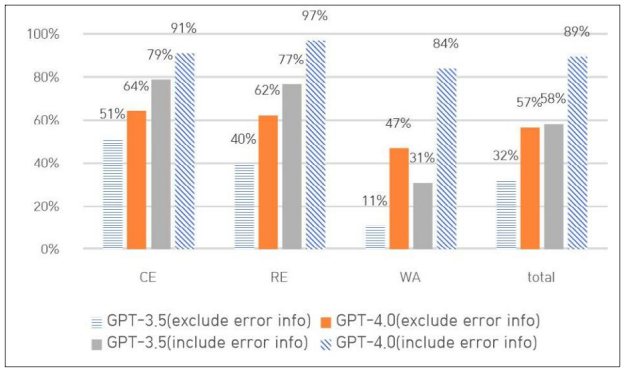

Fig. 3. Descriptive statistics results

Table 10은 GPT-3.5(오답 정보 불포함), GPT-4.0(오답 정보 불포함), GPT-3.5(오답 정보 포함), GPT-4.0(오답 정보 포함)에서 생성된 결과이다. 프롬프트는 Table 4와 Table 6의 내용으로 구성하였으며 제출된 코드의 결과는 Runtime Error이다. 에러가 발생한 사유는 "1000 2000"의 문자열을 띄어쓰기로 분리하지 않고 바로 자료형 변환을 시도하여 발생하였다. 학습자가 제출한 코드로 피드백을 생성하였을 때, 오답 정보가 제공되지 않은 경우에 모두 비정상 피드백으로 판정되었으며, 오답 정보가 제공된 경우에는 모두 정상 피드백으로 판정되었다.

Table 11. Example of feedback

submit code	A = int(input("첫 번째 정수 A를 입력하세요: ")) B = int(input("두 번째 정수 B를 입력하세요: ")) result = A + B print("두 정수의 합은:", result)
GPT-3.5 (exclude error info)	input() 함수를 사용하여 사용자 입력을 받을 때 괄호 안에 문자열을 넣어줘야 합니다.
GPT-4.0 (exclude error info)	입력 함수를 사용하지 않고 코드를 실행할 때 입력 값을 미리 제공해야 하는 경우, 코드 수정 없이 스크립트를 실행하는 환경이나 테스트 코드 내에서 입력 값을 설정하는 방법을 고려해야 합니다.
GPT-3.5 (include error info)	오류 메시지에는 'invalid literal for int()'이라는 부분이 나와 있습니다. 이는 int() 함수가 잘못된 리터럴 값을 받았다는 의미입니다. input() 함수로 입력받은 값이 '1000 2000'와 같이 공백으로 구분된 문자열이기 때문에 int() 함수가 정수로 변환하지 못하는 것입니다. 따라서, 입력을 분리하고 정수로 변환하는 작업이 필요합니다.
GPT-4.0 (include error info)	입력받는 방식을 확인하세요. 띄어쓰기로 구분된 여러 값을 한 번에 처리할 수 있도록 입력 함수와 문자열 분할 방법을 조정해야 합니다.

2. Result for normal feedback about generative AI

생성형 인공지능에 따른 정상 피드백 차이를 확인하기 위해 교차분석을 실시한 결과는 Table 12과 같다. 생성형 인공지능(χ^2 = 139.222, p=.000)에 따라서 정상피드백 여부에 유의미한 차이를 보였다. GPT-4.0(73%)이 GPT-3.5(45%)에 비해 정상 피드백을 생성하는 비율이 높게 나타났다.

Table 12. Chi-square test result for normal feedback about generative AI

category		normal	abnormal	total	χ^2 (p)
GPT3.5	n	393	477	870	139.222 (.000***)
	%	45	55	100	
GPT4.0	n	635	235	870	
	%	73	27	100	

***p < .001

오답 정보가 제공되었을 때, 생성형 인공지능에 따른 정상피드백 차이를 확인하기 위해 교차분석을 실시한 결과는 Table 13와 같다. 생성형 인공지능(χ^2 = 108.630, p=.000)에 따라서 정상피드백 여부에 유의미한 차이를 보였다. GPT-4.0(89%)이 GPT-3.5(58%)에 비해 정상 피드백을 생성하는 비율이 높게 나타났다.

Table 13. Chi-square test result for normal feedback about generative AI(include error info)

category		normal	abnormal	total	χ^2 (p)
GPT3.5	n	254	181	435	108.630 (.000***)
	%	58	42	100	
GPT4.0	n	389	46	435	
	%	89	11	100	

***p < .001

오답 정보가 제공되지 않았을 때, 생성형 인공지능에 따른 정상피드백 차이를 확인하기 위해 교차분석을 실시한 결과는 Table 14과 같다. 생성형 인공지능(χ^2 = 53.344, p=.000)에 따라서 정상피드백 여부에 유의미한 차이를 보였다. GPT-4.0(57%)이 GPT-3.5(32%)에 비해 정상피드백을 생성하는 비율이 높게 나타났다.

Table 14. Chi-square test result for normal feedback about generative AI(exclude error info)

category		normal	abnormal	total	χ^2 (p)
GPT3.5	n	139	296	435	53.344 (.000***)
	%	32	68	100	
GPT4.0	n	246	189	435	
	%	57	43	100	

***p < .001

가설 1을 검정하기 위해 세 번의 교차분석을 실시하였다. 그 결과 생성형 인공지능에 따른 정상피드백 여부에 유의미한 차이를 보였으며, GPT-4.0을 사용하였을 때 정상 피드백을 생성하는 비율이 높게 나타났다. 따라서 가설 1을 채택하였다.

3. Result for normal feedback about prompt

프롬프트에 따른 정상피드백 차이를 확인하기 위해 교차분석을 실시한 결과는 Table 15과 같다. 프롬프트에 오답 정보를 포함 유무(χ^2 = 158.240, p=.000)에 따라서 정상피드백 여부에 유의미한 차이를 보였다. 오답 정보를 제공한 프롬프트(74%)가 오답 정보를 제공하지 않은 프롬프트(44%)에 비해 정상피드백을 생성하는 비율이 높게 나타났다.

Table 15. Chi-square test result for normal feedback about prompt

category		normal	abnormal	total	χ^2 (p)
include err info	n	643	227	870	158.240 (.000***)
	%	74	26	100	
exclude err info	n	385	485	870	
	%	44	56	100	

***p < .001

GPT-3.5에서 프롬프트에 따른 정상피드백 차이를 확인하기 위해 교차분석을 실시한 결과는 Table 16와 같다. 프롬프트에 오답 정보를 포함 유무(χ^2 = 61.377, p=.000)에 따라서 정상피드백 여부에 유의미한 차이를 보였다. 오답 정보를 제공한 프롬프트(58%)가 오답 정보를 제공하지 않은 프롬프트(32%)에 비해 정상 피드백을 생성하는 비율이 높게 나타났다.

Table 16. Chi-square test result for normal feedback about prompt(GPT-3.5)

category		normal	abnormal	total	χ^2 (p)
include err info	n	254	181	435	61.377 (.000***)
	%	58	42	100	
exclude err info	n	139	296	435	
	%	32	68	100	

***p < .001

GPT-4.0에서 프롬프트에 따른 정상피드백 차이를 확인하기 위해 교차분석을 실시한 결과는 Table 17과 같다. 프롬프트에 오답 정보를 포함 유무(χ^2 = 119.220, p=.000)에 따라서 정상피드백 여부에 유의미한 차이를 보였다. 오답 정보를 제공한 프롬프트(89%)가 오답 정보를 제공하지 않은 프롬프트(57%)에 비해 정상 피드백을 생성하는 비율이 높게 나타났다.

Table 17. Chi-square test result for normal feedback about prompt(GPT-4.0)

category		normal	abnormal	total	χ^2 (p)
include err info	n	389	46	435	119.220 (.000***)
	%	89	11	100	
exclude err info	n	246	189	435	
	%	57	43	100	

***p < .001

가설 2를 검정하기 위해 세 번의 교차분석을 실시하였다. 그 결과 프롬프트에 따른 정상 피드백 여부에 대해서 유의미한 차이를 보였으며, 오답 정보를 제공한 프롬프트를 사용하였을 때 정상 피드백을 생성하는 비율이 높게 나타났다. 따라서 가설 2를 채택하였다.

V. Discussion

생성형 인공지능, 오답 정보를 제공한 프롬프트를 사용하여 피드백을 생성한 결과 GPT-4.0을 사용하였을 때, 그리고 오답 정보가 적절히 주어졌을 때 가장 많은 빈도로 정상 피드백을 생성하여 가장 우수한 성능을 나타내었다.

생성형 인공지능의 버전에 따른 정상 피드백 생성의 차이를 비교하였을 때에는 GPT-3.5보다 GPT-4.0이 더 우수하였다(28%). 또한, 프롬프트로 비교하였을 때 오답 정보를 포함한 프롬프트에서 우수한 성능을 나타내었다(30%). 위 결과를 통해 가설 1과 가설 2를 채택하였다.

생성형 인공지능의 버전과 오답 정보를 제공한 프롬프트 중 학습자에게 도움이 되는 피드백 생성에 더 큰 영향을 끼치는 요인을 알아보기 위해 기술 통계 결과를 살펴보면 Compile Error, Runtime Error 의 경우 GPT-4.0에 오답 정보를 제공하지 않은 경우보다 GPT-3.5에 오답 정보를 제공한 경우에 정상 피드백을 생성하는 빈도가 더 높았다. 따라서 적절한 오답 정보를 제공하는 것이 생성형 인공지능 자체의 우수성보다 정상 피드백 생성을 하는데 더욱 큰 영향을 끼치는 것으로 나타난다. 하지만 Wrong Answer의 경우 그 결과가 반대로 나타났는데 이는 Compile Error, Runtime Error가 자체 오류 메시지를 갖는 반면에 Wrong Answer는 오류 메시지를 갖지 못해 나타나는 현상으로 추측된다. 실험을 위해 Wrong Answer의 경우 오류 메시지를 대체하기 위해 Online Judge 서버가 사용하는 input/output 값을 제공하였음에도 불구하고 이러한 결과가 나타난 것은 생성형 인공지능

이 오류의 원인을 올바로 파악하지 못함을 의미한다. 따라서 wrong answer와 같이 문제의 의미를 이해하지 못해서 발생하는 오답에 대해 정상피드백을 생성하기 위한 연구가 필요하다.

VI. Conclusions

본 연구는 프로그래밍 교육에서 맞춤형 교육을 실시하기 위해 자동 튜터링에 필요한 생성형 인공지능이 생성한 피드백을 분석하였다. 이를 위해 초보 학습자가 실제 수업에서 제출한 코드를 사용하였다. 실험을 위해 실제 코드와 오답 정보를 포함한 프롬프트를 작성하였고, 이러한 프롬프트를 사용하여 생성형 인공지능에 피드백을 요청한 결과들을 수집하였다. 본 연구자를 포함한 세 명의 검토진이 피드백의 적절성을 분석하였다. 총 4개의 실험 결과에 대하여 교차검정을 실시한 결과 GPT-4.0을 사용하면서, 적절한 오답 정보를 제공하였을 때, 가장 우수한 성능을 나타내는 것을 확인하였다. 또한 기술 통계 결과를 통해 인공지능 모델 자체의 우수성보다 적절한 오답 정보를 제공하는 것이 더 효과적임을 도출하였다. 특히 이러한 결과는 CE, RE 보다 WA에서 뚜렷하게 드러난다. 따라서 오류 유형에 따라 특화된 프롬프트의 개발이 필요하다.

본 연구의 한계는 다음과 같다. 첫째, 본 연구에서 사용된 데이터는 제한된 인원과 제한된 수업 내용에서 수집된 데이터이다. 따라서 이 데이터 만으로 실험결과를 일반화할 수 없다. 따라서 연구 결과의 보편성을 위해 대규모의 추가 연구가 필요하다.

둘째, 사용된 언어모델은 모두 OpenAI 사의 대표 모델 GPT-4.0인 GPT-3.5이다. 하지만, 최근 OpenAI 사의 GPT-4o뿐 아니라 여러 IT 회사에서 성능이 검증된 여러 언어모델들이 출시된 만큼 추후 연구에서는 다른 모델들을 사용하여 성능을 검증할 필요가 있다.

셋째, 본 연구에서 사용된 프롬프트는 모두 한글로 작성되었다. 하지만 사용된 언어모델은 모두 영문에 최적화되어있으며 동일 내용의 프롬프트일지라도 사용된 언어에 따라서 정상 피드백 생성 확률이 바뀔 수 있다. 따라서, 언어별 비교를 통해 정상 피드백의 비율 변화를 확인할 수 있는 연구가 필요하다.

넷째, 생성형 인공지능이 생성한 피드백이 실제로 학습자가 프로그래밍을 하는 과정에서 긍정적인 영향을 끼치는지 확인해야한다. 본 연구는 검토진이 피드백의 유용성을 판단하였지만, 추후 실제 학습자를 대상으로 생성형 인공지능이 생성한 피드백의 효과성에 대하여 연구를 실시해야 한다.

ACKNOWLEDGEMENT

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. RS-2023-00211436).

REFERENCES

[1] S. C. Kang, S. J. Ahn, Y. H. Sung, Y. S. Jeong, K. Y. A, J. H. Seo and S. Y. Park, "Empirical Data analysis Report On Overseas Software Education Current Status, "Korea Education and Research Information Service.

[2] MINISTRY OF EDUCATION, "the 2022 revised curriculum, " 2022.

[3] MINISTRY OF EDUCATION, "Digital education vision declaration ceremony and academic conference (conference), press realease, "2022.

[4] MINISTRY OF EDUCATION, "Digital-based education innovation plan, "2023.

[5] B. S. Bloom, "The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring," Educational researcher, 13(6), pp. 4-16 1984.

[6] Korean Educational Development Institute, "STATISTICAL YEARBOOK OF EDUCATION, "2023.

[7] S. Kim and Y. Jang, "An Analysis of Factors Influencing High School Students' intention to continue using AI chatbots in Programming Education," The Journal of Korean association of computer education, 26(5), pp. 93-105 2023. DOI:10.32431/kace.2023.26.5.008.

[8] Kim Myunghee, Han Jiwon and Yoo Yung-eui, "A Study on the Effects and Participant Perception of Classes Applying Artificial Intelligence-Based Personalized Learning," Journal of Education & Culture, 29(1) 2023. DOI:10.24159/joec.2023.29.1.137.

[9] Do Jaewoo, Jeongin Eur, Na Yong Jae and Sujin Kim, "A Study of Teachers' Use and Perception of Learning Analytics based Dashboard for Customized Education," The Journal of Korean Teacher Education, 39(4), pp. 261-289. DOI:10.24211/tjkte.2022.39.4.261.

[10] Soohwan Lee and Song Kisang, "Exploration of Domestic Research Trends on Educational Utilization of Generative

Artificial Intelligence," The Journal of Korean association of computer education, 26(6), pp. 15-27 2023. DOI:10.32431/kace.2023.26.6.002.

[11] A. Kurnia, A. Lim and B. Cheang, "Online Judge," Computers & Education, 36(4) 2001.

[12] X. Du, C. Yi, Y. Wei, S. Feng and Z. Gong, "Design of Automata Online Judge," In 2010 2nd International Conference on Information Engineering and Computer Science, pp. 1-4 2010. DOI: 10.1109/ICIECS.2010.5677856

[13] Baekjoon Online Judge, https://www.acmicpc.net/, 2024(Apr 22,).

[14] Jungol, https://www.jungol.co.kr/, 2024(Apr 22,).

[15] W. Chang and S. Kim, "Development and application of algorithm judging system : analysis of effects on programming learning," The Journal of Korean association of computer education, 17(4), pp. 45-57 2014. DOI:http://dx.doi.org/10.32431/kace.2014.17.4.005.

[16] J. Shim and J. M. Chae, "Development of On-line Judge System based onBlock Programming Environment," The Journal of Korean association of computer education, 21(6), pp. 1-10 2018. DOI : 10.32431/kace.2018.21.4.001

[17] S. Jung, "Design of Block Coding Online Judge System," Journal of the Edutainment, 2(1), pp. 57-71 2020. DOI:10.36237/koedus.2.1.57.

[18] E. Sohn and J. Kim, "Implementation of an Algorithmic Trading Problem Evaluation System for Online Programming Courses," KTCP, 29(11) 2023. DOI:10.5626/ktcp.2023.29.11.525.

[19] S. Kim, S. Oh and S. Jeong, "Development and Application of Problem Bank of Problem Solving Programming Using Online Judge System in Data Structure Education," , 21(6), pp. 11-20 2018. DOI : 10.32431/kace.2018.21.4.002

[20] H. Go, J. H. Jeon and Y. Lee, "A Study on the Development of Problem Bank for Programming·Math Convergence Education in Programming Automatic Assessment System, ", 27(2)2023. DOI:10.14352/jkaie.2023.27.2.141.

[21] K. Hur, "Validity Analysis of Python Automatic Scoring Exercise-Problems using Machine Learning Models," , 15(1), pp. 193-198 2023. DOI:10.14702/JPEE.2023.193.

[22] H. W. Kim, H. J. Yun and K. Kim, "A Study on the Intelligent Online Judging System Using User-Based Collaborative Filtering," , 29(1), pp. 273-285 2024. DOI:10.9708/jksci.2024.29.01.273.

[23] H. Mun, S. Kim, J. Kim and Y. Lee, "A Source-code Similarity-based Automatic Tutoring Method for Online Coding Test Service," JOK, 48(9), pp. 1044-1051 2021. DOI:10.5626/jok.2021.48.9.1044.

[24] T. Crow, A. Luxton-Reilly and B. Wuensche, "Intelligent tutoring systems for programming education: a systematic review," Proceedings of the 20th Australasian Computing Education Conference, pp. 53-62 2018. DOI:10.1145/3160489.3160492.

[25] JR Anderson and BJ Reiser, "The LISP Tutor," , 10(4), pp. 159-175 1985.

[26] S. Choi, D. Lee, J. Kim, Y. Jang and H. Kim, "Designing LLM-based Code Reviewing Learning Environment for Programming Education," , 26(5), pp. 1-11 2023. DOI:10.32431/kace.2023.26.5.001.

[27] S. Kim, "Developing Code Generation Prompts for Programming Education with Generative AI," , 26(5) 2023. DOI:10.32431/kace.2023.26.5.009.

[28] S. Lee and K. Song, "Prompt engineering to improve the performance of teaching and learning materials Recommendation of Generative Artificial Intelligence," , 28(8) 2023. DOI:10.9708/jksci.2023.28.08.195.

[29] Code Generation on HumanEval, https://paperswithcode.com/sota/code-generation-on-humaneval , .05.16 2024.

[30] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. Ponde De Oliveira Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, A. Ray, R. Puri, G. Krueger, M. Petrov, H. Khlaaf, G. Sastry, P. Mishkin, B. Chan, S. Gray, N. Ryder, M. Pavlov, A. Power, L. Kaiser, M. Bavarian, C. Winter, P. Tillet, F. P. Such, D. Cummings, M. Plappert, F. Chantzis, E. Barnes, A. Herbert-Voss, W. H. Guss, A. Nichol, A. Paino, N. Tezak, J. Tang, I. Babuschkin, S. Balaji, S. Jain, W. Saunders, C. Hesse, A. N. Carr, J. Leike, J. Achiam, V. Misra, E. Morikawa, A. Radford, M. Knight, M. Brundage, M. Murati, K. Mayer, P. Welinder, B. Mcgrew, D. Amodei, S. Mccandlish, I. Sutskever and W. Zaremba, "Evaluating Large Language Models Trained on Code," arXiv preprint arXiv:2107.03374, 2021.

[31] Qingdao Online Judge, https://github.com/QingdaoU/OnlineJudge.

Authors

Juyeon Lee received the B.S. degree in Mathematical Finance from Gachon University, Korea, in 2021, and received the M.Ed. degree in Computer Education from Korea National University of Education, Korea in 2023. She is interested in computer education.

Seung-Hyun Kim received the Ph.D. degree in computer science from Korea Advanced Institute of Science and Technology (KAIST), Korea. He is currently an Associate Professor in the Department of Computer Education at Korea National University of Education, Korea. He is interested in computer education, information security, and so on.