JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 24, No. 12, pp. 3167-3178, Dec. 2023

Check for updates

한국어 코딩 테스트에서의 인간 대 ChatGPT 3.5 & 4.0 성능 비교 및 평가 체계

최 수 지[1] · 변 혜 원[2*]
[1]성신여자대학교 대학원 미래융합기술공학과 석사과정
[2*]성신여자대학교 AI융합학부 교수

Human Programmers versus ChatGPT 3.5 & 4.0: A Comparison of Coding in Korean

Suzy Choi[1] · Hae-Won Byun[2*]

[1]Master's Course, Department of Convergence Technology Engineering, Sungshin Women's University, Seoul 02844, Korea
[2*]Professor, School of AI Convergence, Sungshin Women's University, Seoul 02844, Korea

[요 약]

본 연구는 대화형 인공지능 모델인 ChatGPT 3.5, ChatGPT4.0과 인간 프로그래머 간의 코딩 문제 해결 능력을 비교 분석하는 것을 목표로 한다. 다양한 난이도와 알고리즘 유형을 포함하는 코딩 문제를 대상으로 정답률, 실행 횟수, 코드 길이, 실행 시간 및 메모리 사용량과 같은 평가 지표를 사용하여 ChatGPT 모델이 생성한 코드와 인간이 작성한 코드를 비교한다. 기존 연구와 달리 본 연구는 특히 한국어 언어에 중점을 두고 생성한 코드 자체의 내용과 품질에 초점을 맞추어, ChatGPT가 생성한 코드와 인간이 작성한 코드 간의 유사성과 차이점 등을 비교 분석한다. 또한, 본 연구는 ChatGPT가 해결하는 코딩 문제 유형이 인간이 해결하는 문제 유형과 어떻게 다른지와 코딩 실패 원인이 서로 다른지에 대해 조사한다. 연구 및 분석에 사용된 코딩 문제 데이터를 포함하여 모든 데이터셋은 공개되어 있으며, 향후 관련 연구가 활발하게 이루어지길 기대한다.

[Abstract]

This study focuses on a comparative analysis between conversational AI models, ChatGPT 3.5 and ChatGPT 4.0, and human programmers based on their ability to solve coding problems. Considering the difficulty levels of coding problems and algorithm types, we compare metrics such as accuracy, execution counts, code length, execution time, and memory usage to assess and compare code generated by the ChatGPT models to that written by human programmers. Unlike previous research, a particular emphasis is placed on the Korean language, focusing on the content and quality of generated code. Furthermore, this study investigates the differences between the types of coding problems addressed by ChatGPT and those tackled by humans, as well as explores the distinct reasons behind coding failures in each case. Datasets used in the research and analysis, including coding problem data, are made publicly available to foster active engagement and promote related research in the future.

색인어 : 인공지능, 챗 지피티, 생성형 AI, 코딩 테스트, 거대언어모델
Keyword : Artificial Intelligence, ChatGPT, Generative AI, Coding Test, Large Language Model

http://dx.doi.org/10.9728/dcs.2023.24.12.3167

Received 07 November 2023; **Revised** 04 December 2023
Accepted 14 December 2023

***Corresponding Author; Hae-Won Byun**
Tel: +82-2-920-7615
E-mail: hyewon@sungshin.ac.kr

I. 서 론

AI가 인간의 생활에 미치는 영향은 이미 헤아릴 수 없을 정도로 광범위해졌다. 그 중에서도 프로그래밍 분야는 AI의 적용 가능성이 매우 높은 분야 중 하나이다. 그러나 AI가 해결할 수 있는 문제의 범위와 한계에 대한 체계적인 연구는 아직 미비한 상태이다. 본 연구는 이러한 문제 인식을 바탕으로, 인공지능, 특히 대화형 인공지능 챗봇인 ChatGPT의 코딩 문제 해결 능력을 측정하고 그 성능을 분석하는 것을 목표로 한다.

ChatGPT는 최근 화제가 되고 있는 대화형 인공지능 챗봇으로서 OpenAI에서 개발한 대형 언어모델(LLM)이다. GPT 모델을 기반으로 만들어진 ChatGPT는 다양한 자연어 처리 작업을 수행하는 능력으로 많은 관심을 받고 있다. 수학, 의학 등 다양한 분야에서 ChatGPT의 문제해결 능력을 평가하는 연구가 진행되고 있다[1]-[10]. 프로그래밍 분야에서도 ChatGPT의 코드 생성, 코드 요약, 오류 수정, 코드 최적화 등의 다양한 측면에서 평가한 연구 결과가 제시되고 있다[11]-[15]. 기존 연구는 코드 생성부터 디버깅, 최적화에 이르기까지 전반적인 프로그래밍 관점에서 ChatGPT 코딩 능력 평가에 초점을 맞추고 있다. 이에 ChatGPT가 생성한 코드 자체에 집중하여 코드의 실행 여부, 코드 길이, 시간 복잡도 등 다각도로 생성한 코드의 성능을 평가할 필요가 있다. 또한, 기존 연구는 영어 코딩 문제를 대상으로 하고 있어서, 한국어 코딩 문제를 대상으로 문제 이해력과 함께 코딩 생성 능력을 분석할 필요가 있다.

다양한 분야에서 ChatGPT 3.5, 4.0과 인간을 비교하는 시도에 착안하여[16]-[18], 본 논문에서는 ChatGPT 3.5, ChatGPT 4.0, 그리고 인간의 3개 주체를 대상으로 코딩 문제해결 능력을 비교하고자 한다. 딥러닝 모델의 학습 방법과 학습 데이터에 따라 ChatGPT 3.5와 4.0의 능력에 차이가 있을 수 있으므로 각 모델의 코딩 능력과 인간의 코딩 능력을 비교한다. ChatGPT가 생성한 코드와 인간이 작성한 코드의 실행 여부, 정답율, 성공하기까지의 실행 횟수, 실행 시간, 메모리 사용량 등을 측정하여 생성 코드의 효율성과 실용성을 분석한다. 코딩 테스트 사이트에서 수준별 문제를 수집하여 ChatGPT가 코딩 테스트에서 어떤 유형의 알고리즘과 문제 유형에 유용한지, 그리고 어떤 종류의 오류에 취약한지를 분석한다. 이를 통해 알고리즘을 공부하는 사람들이 ChatGPT를 교육 도구로서 어떤 수준까지 활용할 수 있는지에 대한 기준을 제시한다.

본 연구의 기여는 다음과 같다.

1. 코딩 테스트 분야에서 GPT-4.0과 이전 버전인 GPT-3.5의 코딩 생성 능력을 참가자와 비교하여 다각도로 분석한다.
2. 한국어로 작성된 코딩 문제를 대상으로 GPT3.5와 GPT4.0의 코딩 문제해결 능력을 분석한다.
3. 실험에 사용된 문제 데이터와 GPT3.5와 GPT4.0이 생성한 코드를 데이터셋으로 공개하여 향후 후속 연구에 기여한다.
4. 데이터 수집과 실험 과정의 대부분을 자동화하여 실험 시간을 단축시켰다.
5. 표, 그래프 및 그림 등을 포함하는 문제도 텍스트로 변환하여 코딩 문제 데이터의 범위를 확장하였다.
6. 코딩을 공부하는 사람들이 ChatGPT를 어느 수준까지 활용할 수 있는지에 대한 기준을 제시하며 교육 분야에서의 AI 기술의 활용 가능성을 탐색한다.

II. 관련 연구

ChatGPT는 다양한 학문 분야에서 적용되어 그 효용성을 입증하고 있다. 의학, 수학, 방사선학 등의 영역에서 이루어진 연구들[1]-[7]은 이를 뒷받침한다. Frieder 등은 ChatGPT의 수학적 해결 능력을 분석했으며[8], 기본 수준의 수학 문제에 대해서는 우수한 성능을 보이지만, 복잡하거나 응용 수학에 관한 질문에 대해서는 정확성과 완전성에서 한계가 있음을 발견하였다. Kung 등은 ChatGPT의 의료 교육 분야 활용 가능성을 조사하였다[9]. 3가지 수준의 문제들을 3종류의 입력 프롬프팅을 통해 ChatGPT에게 질의한 결과, 전반적으로 높은 정답률을 보이며 의학 시험에 합격할 만한 수준을 갖추었다고 평가하였다. 또한 구체적이고 일관된 데이터가 제공된 분야에서 특히 뛰어난 성능을 보였다고 보고하고 있다. Lo는 ChatGPT가 교육 분야에 미치는 영향을 검토하였다[10]. 다양한 영역에서 ChatGPT가 유용하게 사용될 수 있지만 가짜 정보 및 표절과 같은 문제를 해결하기 위해 교사들의 지속적인 노력이 요구된다고 하였다.

프로그래밍 분야에서도 ChatGPT의 성능을 조사하는 연구들이 발표되었다. Biswas는 ChatGPT를 개발자를 도와주는 도구로서의 활용 가능성을 평가하고, 코드 생성, 코드 요약, 프로그래밍 과제 해결 등의 작업에 ChatGPT를 활용하는 방법을 탐구하였다[11]. 코드 완성, 수정, 예측, 오류 수정, 최적화, 문서 생성, 챗봇 개발, 텍스트-코드 생성 및 기술 쿼리 응답 등의 부분에서 ChatGPT를 활용할 수 있으며, 이를 통해 생성 코드의 효율성과 정확성을 향상시킬 수 있다고 평가하였다. Surameery와 Shakor는 ChatGPT가 프로그래밍 오류 해결에 도움을 주는 면에서 특히 우수한 능력을 가지고 있는 점을 강조하였다[12]. 오류 예측, 설명, 및 오류 해결에 ChatGPT 활용이 가능하며, 이를 통해 소프트웨어 개발의 효율성과 정확성을 향상시킬 수 있다고 평가하였다. Kashefi와 Mukerji는 ChatGPT의 수학적 알고리즘 구현 능력을 평가하고, 프로그래밍 언어로 알고리즘을 생성하거나 디버깅하며, 누락된 코드를 완성하거나 다른 프로그래밍 언어로 코드

를 재작성하는 등 다양한 작업에서의 ChatGPT의 활용 가능성을 논의하였다[13].

Chen 등은 ChatGPT가 프로그래밍에 도움을 줄 수 있도록 GPTutor 시스템을 개발하고 성능을 평가하였다[14]. GPTutor 시스템은 기존의 ChatGPT와 GitHub Copilot과 비교하여 간결하고 정확한 설명을 제공하며 컴퓨터 과학 교육에 편리하고 개인화된 도움을 줄 수 있을 것으로 전망하였다. Tian 등은 ChatGPT가 프로그래밍 분야에서 도우미로서 얼마나 활용될 수 있는지를 종합적으로 평가하였다[15]. 코드 생성, 수정, 설명에 대한 검증을 실시한 결과, 생성 측면에서는 전반적으로 높은 정확성을 보이지만 널리 알려지지 않은 문제에는 어려움이 있고, 수정 측면에서는 기존 벤치마크와 경쟁력 있는 결과를 보여준다. 코드 설명 측면에서는 기대에 못 미치는 결과를 얻었다. 종합적으로 ChatGPT가 프로그래밍 도우미로서 도움을 줄 수 있지만 개선이 필요하다고 평가된다. 이에 본 연구는 Tian 등의 연구 중 코드 생성 부분에 중점을 두어 ChatGPT의 성능을 더 깊이 탐구하고자 한다.

III. 데이터셋

본 연구에서는 백준 알고리즘 사이트에서[19] 제공하는 다양한 자료구조와 알고리즘 문제들을 실험에 활용한다. 이 사이트는 국제적인 프로그래밍 대회인 ACM-ICPC의 문제를 포함하여 광범위한 알고리즘 문제를 제공하며, 온라인 저지 시스템을 통해 사용자들이 문제를 풀고 그 결과를 즉시 확인할 수 있는 환경을 제공한다. 대부분의 문제는 한글로 제공된다.

주요 문제 유형은 간단한 구현, 그래프, 수학, 동적계획법, 탐색, 문자열, 브루트포스, 시뮬레이션, 백트래킹, 그리디 등이 있다. 이 사이트는 다양한 프로그래밍 언어(C, C++, Java, Python 등)를 지원하며 문제 난이도와 알고리즘에 따라 분류된 문제를 제공하고 있다. 문제 난이도는 총 6개의 단계(브론즈, 실버, 골드, 플래티넘, 다이아몬드, 루비)로 구성되며, 각 단계는 다시 5개의 세부 단계로 구성된다. 실험을 위해서 각 세부 단계에서 10개의 문제를 선택하여 총 300개의 문제(6개 단계 x 5개 세부 단계 x 10 문제)를 선정하였다. 문제 선택 시, 난이도가 높은 다이아몬드와 루비 단계 문제는 낮은 정답률과 데이터 부족으로 인해 제외되었다. 결과적으로, 브론즈, 실버, 골드, 플래티넘 단계에서 총 200개의 문제를 사용했다.

ChatGPT는 아직까지 이미지 입력을 처리할 수 없기 때문에 이미지를 활용하는 문제는 이미지를 설명하는 텍스트로 변환하였다. 설명이 어려운 이미지가 포함된 문제는 데이터셋에서 제외하였다. 이외에도 프로그래밍 언어를 지원하지 않는 등 적합하지 않은 데이터도 제외하였다. 실험에서 사용한 데이터셋은 문제 데이터, 참가자 문제 풀이 데이터, ChatGPT 문제 풀이 데이터로 구성된다. 표 1은 실험에서 수집한 문제 데이터 목록이다. 문제 데이터는 문제 내용, 입력 및 출력 조건, 입력 및 출력 예시, 시간 제한, 메모리 크기 제한 및 알고리즘 종류로 구성된다. 이 데이터는 ChatGPT에게 질의할 입력 프롬프트를 생성할 때 사용된다.

표 2는 실험에서 수집한 참가자의 문제 풀이 데이터 목록으로 참가자들이 문제를 해결한 결과 데이터이다. 참가자의 문제 풀이 데이터는 백준 알고리즘 사이트[19]에서 실제로 답변을 제출한 사람들의 데이터를 수집하였다. 제출된 정답 코드에서 순위를 기준으로 중간값(median) 200개의 샘플을 선택하여 분석하였다. 이는 데이터의 중앙값을 기준으로 하여 전체 데이터의 대표성을 확보하는 동시에, 극단적인 값들이 결과에 미치는 영향을 최소화하기 위함이다. 데이터는 정답률, 평균 시도 횟수, 오류 유형별 인원수, 평균 실행시간, 평균 메모리 크기, 평균 코드 길이로 구성된다.

표 1. 문제 결과 데이터 수집 예시
Table 1. Example of problem results data collection

Data	Example
Question	Write a program that takes two integers, A and B, as input and outputs A minus B. Input: A and B are given on the first line. (0 < A, B < 10)
Input and output specifications	Input: A and B are given on the first line. (0 < A, B < 10) Output: Output A minus B on the first line.
Input and output examples	Example Input 1: 3 2 Example Output 1: 1
Memory constraints	128 MB
Time constraints	2 초 / 2 sec
Algorithm classification	Mathematics Implementation Arithmetic Operations

표 2. 참가자 결과 데이터 수집 예시
Table 2. Example of participants result data collection

Data	Example
Accuracy rate	70.314%
Average number of attempts	1.4222 times
Number of participants by error type	Output format: 299 Wrong answer: 28325 Time limit exceeded: 147 Memory limit exceeded: 85 Output limit exceeded: 81 Runtime error: 27292 Compilation error: 58139
Average code length	69B
Average execution time	108ms
Average memory usage	114,107KB

표 3은 알고리즘 사이트에 ChatGPT 코드를 제출하여 채점한 결과와 수집한 데이터 목록 및 예시를 보여 준다. 채점 결과는 코드의 정답 여부와 오답인 경우에 오답 사유를 명시한다. “틀렸습니다”와 같은 단순한 정답 오류뿐만 아니라, 출력 형식 오류, 시간 초과 오류, 메모리 크기 초과 오류, 출력 초과 오류, 런타임 오류, 컴파일 오류 등 다양한 형태의 오류를 구분하여 기록한다. 또한, 참가자 데이터와의 비교 분석을

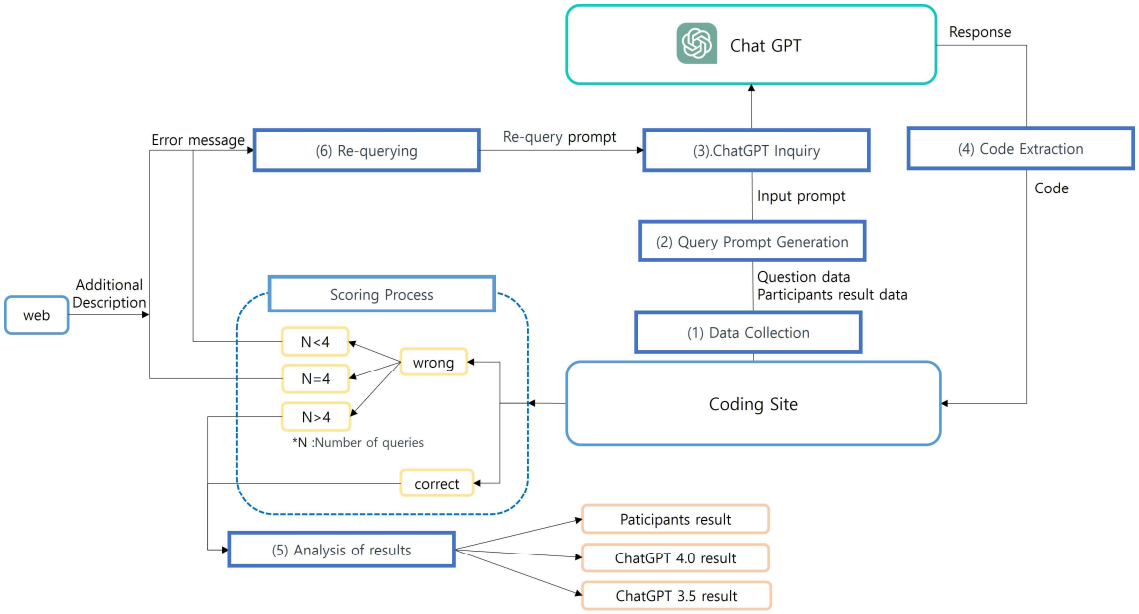


그림 1. 실험 구성도
Fig. 1. Experimental setup

위해 ChatGPT가 생성한 코드 길이, 실행 시간, 메모리 사용량 데이터도 함께 수집한다.

본 실험은 실험에 사용한 문제 데이터와 생성한 코드를 공개함으로써 향후 후속 연구에 기여할 수 있는 기반을 마련한다[20]. 문제 데이터의 저작권은 백준 알고리즘 사이트[19]에 있으므로, 각 문항의 웹 페이지 링크로 대신한다. 이를 통해 누구나 문제와 참가자 데이터를 확인할 수 있다. 데이터셋은 총 4회 질문의 답변 코드와 채점 결과로 구성된다. 채점 결과는 T, F, N, X로 이루어지며, T는 정답, F는 오답, N은 추가 설명을 더하여 얻은 정답, X는 사용 불가한 문제를 의미한다. 이 데이터셋을 활용하여 수준별 코딩 테스트 문제를 활용하고 GPT 코딩 성능을 분석할 수 있을 것으로 기대한다.

표 3. ChatGPT 결과 데이터 수집 예시
Table 3. Example of ChatGPT result data collection

Data	Example
Scoring Results	Correct Incorrect,Output format, Time limit exceeded. Memory limit exceeded, Output limit exceeded, Runtime error, Compilation error. one from these.
Code length	44B
Execution time	116ms
Memory usage	113,112KB

*This is actual data examples and the data is in Korean.

IV. 실험방법

이 절에서는 실험방법의 전체적인 개요를 소개한다. 그림 1을 보면, 코딩 플랫폼에서 수집한 코딩 문제 데이터를 기반으로 ChatGPT용 질의 프롬프트를 생성하고 이를 ChatGPT 모델에 전달하여 해답 코드를 생성하고, 코딩 사이트에서 생성 코드를 실행, 채점한다. 채점 결과 오답인 경우, 실제 참가자와 동일하게 2회의 추가 기회를 제공하기 위하여 해당 오류 코드를 포함하여 추가적인 재질의 과정을 2번 더 반복 수행한다. 2회 수행 이후에도 해답 코드를 생성하지 못하면, 웹에서 문제 해결에 필요한 추가 설명을 수집하여 4번째 재질의 과정을 한 번 더 마지막으로 수행한다. 이후, 정답 또는 오답 결과를 수집하여 실험 결과를 분석한다. 실험 절차의 대부분을 자동화하여 수동 작업을 최소화하고 실험 시간을 단축시키고자 하였다.

1) 데이터 수집

파이썬 기반의 크롤링 기술을 사용하여 데이터를 수집하였다. Selenium[21]을 이용하여 웹 브라우저를 자동화하고 Beautiful Soup[22]을 통해 필요한 정보를 추출한다. 이 과정은 자동화를 통해 난이도별 문제 목록과 세부 데이터를 효율적으로 수집한다.

2) ChatGPT 질의 프롬프트 생성

수집한 문제 데이터를 기반으로 ChatGPT가 문제를 이해하고 코드를 생성할 수 있는 질의 프롬프트를 생성한다. 질의 프롬프트는 “아래 문제를 푸는 파이썬 코드 알려줘” 라는 문장을 포함하여 코딩 플랫폼에서 추출한 문제 및 입력과 출력에 관한 설명으로 구성된다. 이와 더불어 코딩 사이트에서 수집한 예시 입력과 예시 출력 그리고 시간 제한 및 메모리 제한 조건도 명시한다. 여러 개의 예시 입력과 예시 출력이 존재하는 경우, 각 예시들을 식별하기 위해 번호를 부여하고 이

를 질의 프롬프트에 포함시킨다. 그림 2는 생성한 질의 프롬프트의 예시이다.

자동화가 어려운 일부 문제를 제외하고, 대부분의 문제에 대해서 질의 프롬프트를 생성하는 과정을 자동화하였다. 표와 그림이 포함된 문제의 경우 추가적인 프롬프트 엔지니어링을 그림 3과 같이 수행한다. 표는 HTML 테이블 형식으로 변환하고 그림은 텍스트 설명으로 변환한다. 다만, 텍스트로 표현하기 어려운 그림이 포함된 일부 문제들은 실험 대상에서 제외하였다.

아래 문제를 푸는 파이썬 코드 알려줘
[문제] N을 입력받은 뒤, 구구단 N단을 출력하는 프로그램을 작성하시오. 출력 형식에 맞춰서 출력하면 된다. 시간 제한은 1 초이고 메모리 제한은 128 MB입니다.
[입력] 첫째 줄에 N이 주어진다. N은 1보다 크거나 같고 9보다 작거나 같다.
[출력] 출력 형식과 같게 N*1부터 N*9까지 출력한다.
[예시 입력1] 2
[예시 출력1]
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10
2 * 6 = 12
2 * 7 = 14
2 * 8 = 16
2 * 9 = 18

*This is actual data examples and the data is in Korean.

그림 2. 입력 프롬프트 예시
Fig. 2. Example input prompt

3) ChatGPT 질의

질의 프롬프트가 준비되면, ChatGPT 모델에 질의 프롬프트를 입력하여 답변 데이터를 수집한다. GPT3.5의 경우, OpenAI에서 제공하는 API를 사용하여 질의한다. GPT4.0의 경우, 아직 API가 제공되지 않기 때문에 웹 페이지에서 직접 수동으로 질의한다. 이전 질의 결과가 현재 질문에 영향을 미칠 수 있는 가능성을 배제하기 위해서 각 질의는 별도의 세션에서 수행하였다. 그림 4는 GPT3.5 API를 이용하여 질문한 예시 프롬프트이다. 질문 시, GPT의 '역할(role)'을 '사용자(user)'로 지정하고, '내용(content)'에는 준비된 질의 프롬프트를 입력한다.

"role": "user,"
"content": "아래 문제를 푸는 파이썬 코드 알려줘\n\n\n문제 : 수열 A가 주어졌을 때, 가장 긴 증가하는\xa0부분\xa0수열을 구하는 프로그램을 작성하시오.\n입력 : 첫째 줄에 수열 A의 크기 N (1 ≤ N ≤ 1,000,000)이 주어진다. 둘째 줄에는 수열 A를 이루고 있는 Ai가 주어진다. (-1,000,000,000 ≤ Ai ≤ 1,000,000,000)\n출력 : 첫째 줄에 수열 A의 가장 긴 증가하는 부분 수열의 길이를 출력한다. 둘째 줄에는 정답이 될 수 있는 가장 긴 증가하는 부분 수열을 출력한다.\n예시 입력1 : 6\n10 20 10 30 20 50\n\n예시 출력1 : 4\n10 20 30 50\n\n\n시간 제한은 3 초이고, 메모리 제한은 512 MB입니다"

*This is actual data examples and the data is in Korean.

그림 4. ChatGPT3.5 API 입력 예시
Fig. 4. Example Input for ChatGPT3.5 API

4) ChatGPT 답변 코드 추출

이 단계에서는 GPT 모델이 생성한 답변에서 코드 부분만을 추출한다. GPT3.5의 경우 API를 사용하여 질의하였기 때문에 그림 5와 같이 JSON 형식의 답변을 얻는다. JSON형식의 답변 중에서 'content' 부분이 실제 답변이며, 답변 중에서 문제 풀이 방법 설명, 코드 설명, 알고리즘 설명 등을 제외하고 코드 부분만을 선별하여 추출한다. 만약 답변이 코드만으로 이루어져 있는 경우에는 전체 답변을 사용한다.

'choices': [{
'finish_reason': 'stop',
'index': 0,
'message': {'
'content': 'n = int(input())\n\nfor i in range(1, 10):\n
print(n, "*", i, "=", n*i)', 'role': 'assistant'}}],
'created': 1680134939,
'id': 'chatcmpl-6za8Z2A9aK0Eiu4ZGHJtN0kc10Kd9',
'model': 'gpt-3.5-turbo-0301',
'object': 'chat.completion',
'usage': {
'completion_tokens': 26,
'prompt_tokens': 254,
'total_tokens': 280}

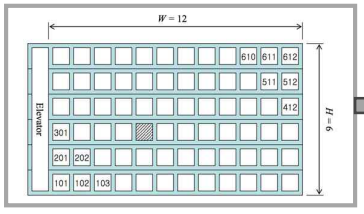
그림 5. ChatGPT3.5의 API 답변 예시
Fig. 5. Example response from ChatGPT3.5 API

a. 표를 html형식으로 변환

1/1	1/2	1/3	1/4	1/5	...
2/1	2/2	2/3	2/4	...	...
3/1	3/2	3/3	...	...	...
4/1	4/2	...	...	...	...
5/1	...	...	...	...	...
...	...	...	...	...	...

```
<table class="table table-bordered" style="width:30%">
  <tbody>
    <tr>
      <td style="width:5%">1/1</td>
      <td style="width:5%">1/2</td>
      <td style="width:5%">1/3</td>
      <td style="width:5%">1/4</td>
      <td style="width:5%">1/5</td>
      <td style="width:5%">&hellip;</td>
    </tr>
    <tr>
      <td>2/1</td>
      <td>2/2</td>
      <td>2/3</td>
      <td>2/4</td>
      <td>&hellip;</td>
      <td>&hellip;</td>
    </tr>
    <tr>
      <td>3/1</td>
      <td>3/2</td>
      <td>3/3</td>
      <td>&hellip;</td>
      <td>&hellip;</td>
      <td>&hellip;</td>
    </tr>
    <tr>
      <td>4/1</td>
      <td>4/2</td>
      <td>&hellip;</td>
      <td>&hellip;</td>
      <td>&hellip;</td>
      <td>&hellip;</td>
    </tr>
    <tr>
      <td>5/1</td>
      <td>&hellip;</td>
      <td>&hellip;</td>
      <td>&hellip;</td>
      <td>&hellip;</td>
      <td>&hellip;</td>
    </tr>
    ody>
</table>
```

b. 그림을 텍스트로 설명

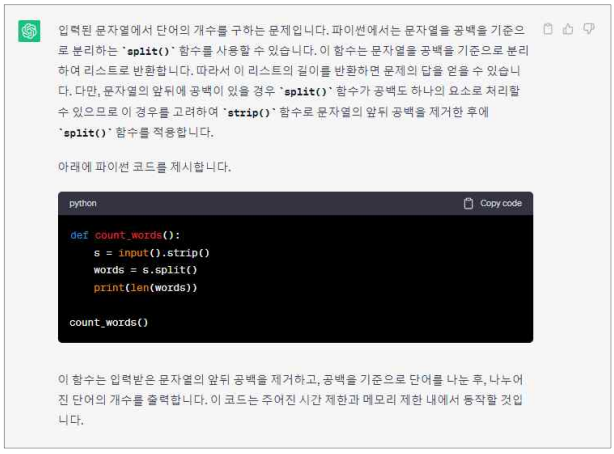


즉, 방이 3층이고 왼쪽으로부터 5번째이면 305호가 된다.

*This is actual data examples and the data is in Korean.

그림 3. 프롬프트 엔지니어링 예시
Fig. 3. Prompt engineering example

'role'이 'assistant'로 지정된 부분은 재질의 시 ChatGPT의 역할인 "role"을 지정하는데 사용되며, 또한 재질의 과정에서 질의 문맥을 유지하기 위해 필요한 "id" 데이터를 추출하여 저장한다. GPT4.0 모델의 경우, 웹 페이지에서 얻은 답변 코드에서 코드 부분만을 추출한다. GPT4.0의 답변은 그림 6과 같다. GPT4.0의 답변은 코드를 포함하지 않은 경우가 가끔 발생한다. 첫째, 문제 자체가 어려워서 답변을 생성하지 못한 경우, 둘째, 문제 해결 방법을 설명하면서 코드는 제공하지 않는 경우, 셋째, GPT 답변 내용없이 답변 ID만을 제공한 경우가 있다. 실험에서는 세 가지 경우를 별도로 분류하여 처리하였다.



*This is actual data examples and the data is in Korean.

그림 6. ChatGPT4.0의 답변 예시

Fig. 6. Example response from ChatGPT4.0.

5) 채점 결과 데이터 수집

ChatGPT가 생성한 코드를 채점하기 위해서 해당 코드를 온라인 저지 사이트에 제출한다. 이 과정에서 코드의 정답 여부를 평가하며 코드가 정답이 아닌 경우에는 오류 코드를 기록한다. 오류의 종류로는 "틀렸습니다"와 같은 일반적인 오류 외에도 출력 형식 오류, 시간 초과 오류, 메모리 크기 초과 오류, 출력 초과 오류, 런타임 오류, 컴파일 오류 등이 있다.

코드 제출 시에는 pypy3 언어를 사용한다. 이는 Python으로 작성한 코드를 제출하는 경우 시간 초과 오류가 자주 발생할 수 있어서 이를 방지하기 위함이다. 코드는 Python으로 작성되었지만 성능 향상을 위해 pypy3로 제출한 것이므로, 실제 참가자가 작성한 코드의 데이터 수집 시에도 동일한 조건을 적용한다. 따라서 평균 실행시간과 평균 메모리 사용량은 제출된 언어 기준인 pypy3로, 평균 코드 길이는 코드 기준인 python으로 정보를 수집하였다.

6) ChatGPT 재질의

실제로 모든 코드를 한 번에 완벽하게 작성하는 일은 어려운 일이므로 코드의 오류를 발견하고 수정하는 작업은 중요한 과정이다. ChatGPT 모델의 경우에도 오류가 발생한 이유를 설명하고 문제를 해결하기 위해 추가 질의를 수행하는 재질의 과정을 실험적으로 시도하였다. 재질의 과정에서 ChatGPT는 코드를 수정하여 정답 코드를 생성하는 경우가 많았다. GPT3.5의 경우 API를 통해 다시 질의하면서 이전 질문의 문맥을 유지하기 위해 답변 ID를 사용한다. 정답 코드를 생성하지 못하는 경우, ChatGPT에게 다시 질의하여 생성된 답변 코드를 채점하고 정답이 틀린 문제에 대해서 최대 3회까지 재질의한다.

GPT가 3번의 시도 내에 정답을 찾지 못하는 경우, 해당 문제는 해결할 수 없는 것으로 간주하고, 정답 코드에 인터넷 검색으로 찾은 설명을 추가하여 재질의 과정을 수행한다. 표 4에서 볼 수 있듯이 인터넷 검색으로 얻은 추가 설명은 6개의 유형으로 분류된다. 이 중에서 의사 코드(pseudo code)나 답변 코드를 직접 포함하는 문서와 답변과 관계없는 설명만 있는 문서를 제외하고, 코드를 포함하지 않는 일반적인 문제 해결 방법, 절차적 설명, 그리고 사용하는 알고리즘 설명의 우선순위로 추가 설명을 선정한다.

추가 설명을 제공하더라도 문제를 해결하지 못한 경우는 실패로 간주하였다. 표 5는 각 질의 단계에서 사용한 질의 프롬프트를 보여준다.

표 4. GPT에게 제공한 설명글의 분류

Table 4. The categorization of descriptive texts provided to GPT

Solution search on the Internet	
Explanation type	Answer code
No explanation	O
Problem statement	O
Procedural explanation	O
	X
Procedural explanation + Pseudocode	O
	X
Algorithm explanation	O
	X
Solution approach	O
	X

표 5. 질의 횟수별 질의 프롬프트

Table 5. Query prompt by number of queries

Queries	Input data	Query prompt example
First query	Initial generated input prompt	Please provide a Python code to solve the following problem: (The problem statement is omitted). The time limit is 1 second, and the memory limit is 128 MB.
Second query	Error message	The code in the previous answer is incorrect. Please provide the code again.
Third query	Error message	The code in the previous answer resulted in a runtime error. Please provide the code again.
Fourth query	Error message, additional explanation	The code in the response is incorrect. (Explanation omitted) Please refer to the explanation and solve it again.

V. 실험결과

본 실험에서는 코딩 사이트에 있는 프로그래밍 문제들에 대해서 GPT3.5와 GPT4.0, 참여자들의 코드 생성 능력을 평가한다. 실험을 위해 총 200개 문제 중 186문제를 결과 분석에 사용하였다.

그림 7은 ChatGPT3.5, ChatGPT4.0과 참가자의 정답률을 비교하는 막대 그래프이다. ChatGPT는 난이도가 낮은 2단계 문제(브론즈, 실버 수준)에 대해서는 100% 정답률을 보여주었다. 이는 인터넷에 존재하는 다량의 브론즈, 실버 수준의 코딩 문제 풀이 데이터를 이용하여 ChatGPT를 학습시킨 결과로 해석된다. 고수준으로 갈수록 ChatGPT의 정답률이 저하되고 다이아몬드 이상 수준의 문제에는 대응하지 못하는 것을 확인할 수 있다. 골드와 골드 수준에서는 ChatGPT4.0이 3.5보다 높은 성능을 보여주고 있다. ChatGPT 대비 참가자의 경우, 중위 표본에 속하는 사람들을 대상으로 비교하여 가장 낮은 수준의 문제 정답률이 52%이고 높은 수준으로 갈수록 정답률은 저하된다. 다이아몬드와 루비 수준의 고수준 문제도 해결하고 있는 점이 ChatGPT와 다른 부분이다.

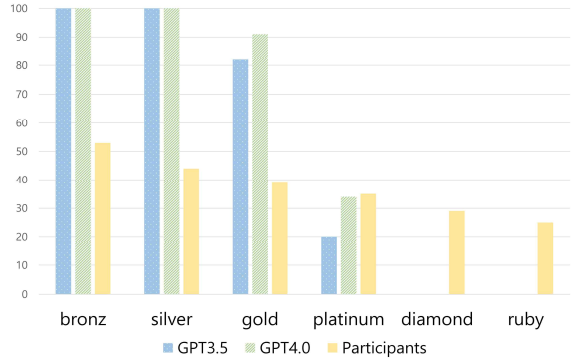


그림 7. GPT3.5, GPT4.0, 참가자 대상 코딩 문제 수준별 정답률

Fig. 7. GPT3.5 and GPT4.0, participant success rates by level

그림 8은 정답을 맞힌 문제에 대해 성공 전 시도 횟수를 보여 준다. 대부분의 문제에서 1차 시도에서 정답을 맞힌 비율이 가장 높았으며, 문제의 난이도가 높아질수록 시도 횟수가 증가하는 것을 확인할 수 있다. 노란 막대는 코드 생성 실패 이후에 추가 설명을 제공하여 정답을 맞힌 문제의 수다. 추가 설명을 활용하여 GPT4.0은 정답을 맞히는데 성공했지만, GPT3.5는 추가 설명을 제공해도 정답을 맞히지 못하였다. 이는 GPT3.5가 한글로 작성된 추가 설명을 읽고 이해하는 능력이 GPT4.0에 비해 부족한 것으로 분석된다.

알고리즘 유형별 코딩 문제에 대한 정답률을 분석하였다. 코딩 사이트에서 모든 유형의 알고리즘을 수집하였지만, 문제 수가 너무 적어서 무의미하다고 판단되는 데이터를 제외하고 가장 많이 사용된 알고리즘 상위 6개만 선정하여 분석하였다.

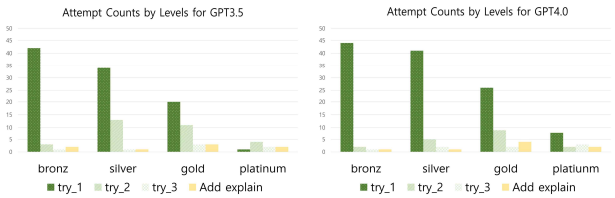


그림 8. 수준별 시도 횟수

Fig. 8. Attempt counts by levels

표 6은 알고리즘 유형별 GPT3.5, GPT4.0의 시도 횟수(try_cnt) 및 정답률(rate) 대비 참가자의 정답률을 비교하고 있다. GPT 3.5, GPT4.0, 그리고 참가자 모두 단순 구현 문제, 그래프 이론, 수학 문제에서 높은 정답률을 보여 주고 있다. 이는 인간이 해결한 코드를 인터넷에서 수집하여 ChatGPT를 훈련시켰기 때문인 것으로 해석된다. 각 알고리즘 유형에 대해 GPT3.5와 4.0의 정답률이 다소 차이를 보이는 점은 흥미로운데, 이는 서로 다른 데이터셋을 이용하여 서로 다른 방법으로 ChatGPT를 학습시켰기 때문으로 판단된다. 그리고 ChatGPT와 비교하여 참가자의 경우 코딩 문제 해결력에 있어서 알고리즘 유형별로 편차가 적은 것을 확인할 수 있다. 참가자는 다양한 알고리즘 분야의 코딩 문제를 해결할 수 있지만, ChatGPT는 알고리즘 유형별 학습 데이터 개수에 영향을 받아서 분야에 의존적으로 코딩 문제 해결력에 다소 큰 편차를 보이고 있는 것으로 분석된다. 시도 횟수(try_cnt)를 분석해 보면, 그래프 이론을 제외하고 전반적으로는 문제가 어려워서 정답률이 낮을수록 시도 횟수가 증가하는 반비례 특성을 확인할 수 있다.

표 6. 알고리즘 분류별 시도횟수와 정답률 비교

Table 6. Attempt counts and accuracy by algorithm category

	GPT3.5		GPT4.0		Participants
Algorithm	try_cnt	Rate(%)	try_cnt	Rate(%)	Rate(%)
Implementation	1.2	85	1.0	77	51
Graph theory	1.5	75	1.3	77	37
Mathematics	1.2	56	1.1	60	42
Dynamic prog.	1.4	31	1.2	27	38
Greedy algorithm	1.7	27	1.3	27	39
Data structure	1.7	25	1.5	21	30

GPT3.5, GPT4.0, 참가자 중에서 어떤 코드가 더 효율적인지 각각 제출한 코드를 분석하였다. 메모리 사용량, 실행 시간, 코드 길이를 기준으로 각 코드의 효율성을 평가하였다. 그림 9는 GPT3.5, GPT4.0 참가자의 코드를 비교한 결과이며 하단의 그래프는 상단의 그래프를 확대한 것이다. 성능 비교는 GPT3.5와 GPT4.0이 모두 맞힌 문제를 대상으로 하고, GPT3.5, GPT4.0, 참가자 모두 심각한 이상치를 갖는 문제 1개는 집계에서 제외하였다. 메모리 사용량 측면에서 보면, ChatGPT는 참가자보다 메모리를 적게 사용하고, GPT4.0의 메모리 사용량이 GPT3.5보다 다소 적은 것을 확인할 수 있

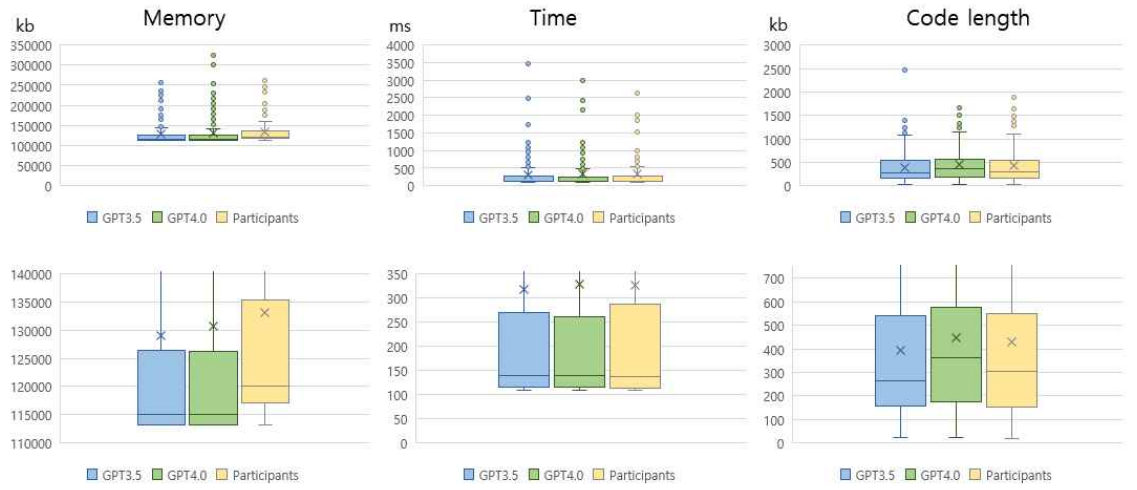


그림 9. GPT3.5, GPT4.0, 참가자의 코드 성능 비교
Fig. 9. Comparison of code performance between GPT3.5, GPT4.0, and participants

a. GPT 3.5

```
import sys

n = int(sys.stdin.readline())
stack = []

for i in range(n):
    command = sys.stdin.readline().split()
    if command[0] == 'push':
        stack.append(int(command[1]))
    elif command[0] == 'pop':
        if not stack:
            print(-1)
        else:
            print(stack.pop())
    elif command[0] == 'size':
        print(len(stack))
    elif command[0] == 'empty':
        if not stack:
            print(1)
        else:
            print(0)
    elif command[0] == 'top':
        if not stack:
            print(-1)
        else:
            print(stack[-1])
```

b. GPT 4.0

```
import sys

class Stack:
    def __init__(self):
        self.stack = []

    def push(self, x):
        self.stack.append(x)

    def pop(self):
        return self.stack.pop() if self.stack else -1

    def size(self):
        return len(self.stack)

    def empty(self):
        return 1 if not self.stack else 0

    def top(self):
        return self.stack[-1] if self.stack else -1

def main():
    n = int(sys.stdin.readline())
    stack = Stack()

    for _ in range(n):
        command = sys.stdin.readline().split()

        if command[0] == "push":
            stack.push(int(command[1]))
        elif command[0] == "pop":
            print(stack.pop())
        elif command[0] == "size":
            print(stack.size())
        elif command[0] == "empty":
            print(stack.empty())
        elif command[0] == "top":
            print(stack.top())

if __name__ == "__main__":
    main()
```

c. Participant1(high score)

```
from sys import stdin,stdout
input = stdin.readline


n=int(input())
l=[]
for _ in range(n):
  cmd=input().rstrip()

  if cmd=="pop":
    stdout.write(l.pop()+"\n") if l else
stdout.write("-1\n")
  elif cmd=="size":
    stdout.write(str(len(l))+"\n")
  elif cmd=="empty":
    stdout.write("0\n" if l else "1\n")
  elif cmd=="top":
    stdout.write(l[-1]+"\n") if l else
stdout.write("-1\n")
  else:
    l.append(cmd.split()[1])
```

d. Participant2(low score)

```
N = int(input())
stack = []
pstack = []
for i in range(N) :
    stackInput = input()
    if "push" in stackInput  :
        push = stackInput.split(" ")
        stack.insert(0, push[1])
    elif stackInput == "pop" :
        if stack == [] :
            pstack.append("-1")
        else :
            pstack.append(stack.pop(0))
    elif stackInput == "size" :
        pstack.append(len(stack))
    elif stackInput == "empty" :
        if stack == [] :
            pstack.append("1")
        else :
            pstack.append("0")
    elif stackInput == "top" :
        if stack == [] :
            pstack.append("-1")
        else :
            pstack.append(stack[0])

for i in pstack :
    print(i)
```

그림 10. GPT3.5와 GPT4.0 그리고 참가자가 생성한 코드 비교
Fig. 10. Comparison between GPT3.5, GPT4.0, and participant generated codes

다. 실행 시간 측면에서도 GPT4.0이 가장 우수한 성능을 보여주고 있고 참가자의 실행시간이 가장 긴 것으로 나타났지만, 전반적으로 근소한 차이이다.

반면, 코드 길이 부분에서 흥미로운 점은 GPT3.5의 코드가 가장 짧고, 참가자, GPT4.0 순으로 GPT3.5의 성능이 가장 우수하다. GPT4.0은 코드 모듈화 및 성능 최적화를 보다 높은 수준으로 적용하기 때문에 코드가 길어진 것으로 해석된다. 그림 10을 보면, GPT3.5와 4.0이 각각 생성한 코드를 비교해 볼 수 있다. GPT3.5 코드는 대부분 절차적으로 구성되어 있는 반면, GPT4.0 코드는 클래스를 이용하여 구조화를 시도한다. 코드 길이 기준으로만 단순 평가하면 GPT3.5의 성능이 좋다고 해석할 수 있지만, 코드의 가독성과 유지보수 측면을 기준으로 평가하면 GPT4.0의 코드가 더 효과적이라고 분석할 수 있다.

GPT가 생성한 코드와 사람이 작성한 코드 내용을 관찰하고 그 효율성을 분석하였다. 그림 10에서 보면, GPT3.5 코드 a는 간단하고 직관적인 코드로 이해하기 쉽고, GPT4.0 코드 b는 객체 지향적 구조로 모듈화되어 가독성이 높고 유지보수가 쉽다. 코드 c와 d는 실제 참가자가 작성한 코드이다. 코딩 사이트에서 참가자들의 코드는 메모리 사용량, 시간, 코드 길이를 기준으로 순위가 매겨지는데, c는 높은 순위에 있는 코드이고 d는 낮은 순위에 있는 코드이다. 코드 c는 표준 입출

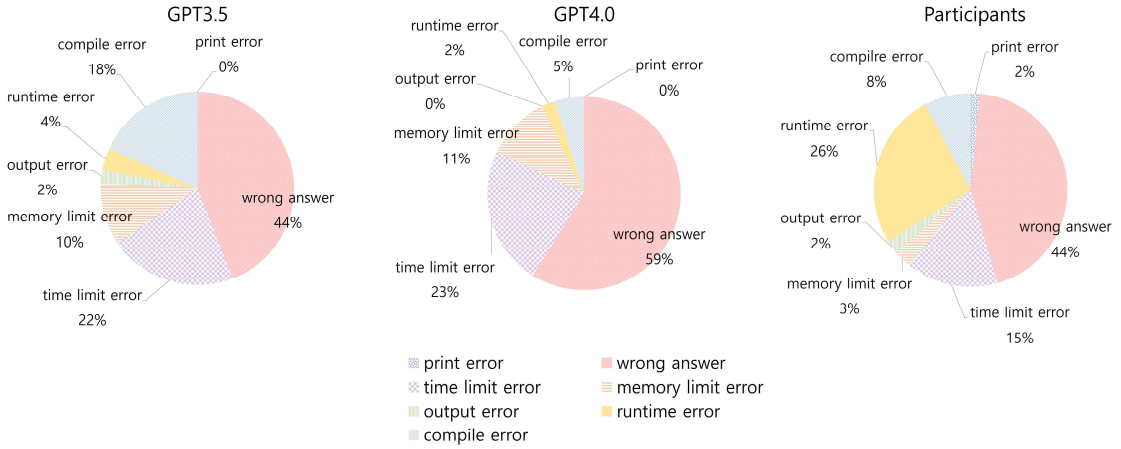


그림 11. GPT3.5,GPT4.과 참가자의 오류 유형 분포
Fig. 11. Comparison of code performance between

력을 활용하여 코드가 간결하여 가독성이 좋고, 코드 d는 리스트의 맨 앞에서 원소를 추가/삭제하는 작업에서 O(n) 시간이 소요되므로 상대적으로 시간 복잡도가 크다. 종합적으로, 코딩 테스트를 위한 간단한 기능은 코드 a와 b가 적합하다. 그러나 재활용성을 고려하고 교육을 목적으로 한다면 코드 c와 같이 객체 지향적인 접근이나 코드의 구조화가 필요하다. 코드 d는 시간 복잡도가 커서 성능 측면에서는 비효율적이다. ChatGPT를 이용한 코드 생성 시, 코드 생성의 목적을 정확하게 표현하는 등 프롬프트 엔지니어링이 중요하다는 것을 시사한다.

표 7에서 GPT3,5, GPT 4.0 및 참가자가 생성한 코드의 상세 내용을 분석한다. 코드 길이, 클래스의 사용 여부, 모듈화, 주석 포함, 그리고 컨테이너의 사용 여부를 포함하는지 등을 비교한다. 대부분의 항목에서 GPT4.0이 가장 우수한 코드를 작성하고 있는 것을 확인할 수 있다. GPT3.5와 참가자는 클래스 구현을 거의 사용하지 않는 반면에 GPT4.0은 일부 코드에서 클래스를 사용하고 있다. 또한, GPT4.0은 대부분의 코드에서 함수를 활용한 모듈화를 구현하였으며 컨테이너 사용 빈도도 상대적으로 높았다. GPT가 생성한 코드는 주석도 포함되어 있다.

표 7. GPT3.5, GPT4.0과 참가자의 코드 비교
Table 7. Code comparison between GPT3.5, GPT4.0, and participants

Algorithm	GPT3.5	GPT4.0	Participant1 (high score)	Participant2 (low score)
Code length	short	moderate	short	long
Class utilization	5%	10%	5%	0%
Function usage	70%	90%	70%	50%
Comment usage	90%	90%	30%	30%
Container usage	10%	30%	10%	10%

그림 11은 GPT3.5, GPT4.0과 참가자들이 문제를 해결하는 코드를 제출했을 때 발생한 오류 유형의 분포 비율을 시각화한 것이다. 오류 유형별로 분석해 보았을 때, GPT의 경우 3.5와 4.0 모두 단순 정답 오류인 '틀렸습니다(wrong answer)'가 가장 높은 비율을 차지했으며 시간 초과 오류가 뒤를 이었다. 반면에 사람이 작성한 코드의 오류 유형을 살펴보면 GPT와 같이 단순 정답 오류의 비율이 가장 높고 런타임 에러가 두 번째로 많은 오류로 나타났다. 특히 런타임 에러는 GPT에 비해 훨씬 높은 비율을 보인다. 세 번째 시간 초과 오류가 많은 것으로 보아 전반적인 오류 비율은 GPT와 비슷하지만, GPT가 런타임 에러에 비교적 강하다고 볼 수 있다.

GPT3.5의 경우, 22개의 문제에서 코드가 포함되지 않은 답변을 제공하였다. 단순히 알고리즘을 설명하는 문구만 제공하거나, 문제가 어렵다며 코드를 제공하지 않거나, 또는 ChatGPT API가 제공하는 "id" 형식의 문자열을 답변으로 제시하고 있다. "id"만으로는 답변 코드를 추출할 수 없기 때문에 웹 페이지에서 직접 질의하는 재실험을 진행하였다. 웹 페이지 질의 결과, "id"형식의 답변이 제시되지 않았으므로 이 문제는 API 오류로 분석된다. 그러나 웹 페이지에서 재실험한 모든 문항의 결과에서 정답률에 차이가 없었기 때문에 GPT3.5의 API 오류를 감안하더라도 GPT4.0이 더 뛰어난 질문 의도 파악 능력을 보여주고 있다고 할 수 있다.

VI. 결 론

본 연구는 ChatGPT 3.5, ChatGPT 4.0 및 인간 프로그래머 간의 한국어 코딩 테스트에서의 코딩 능력을 비교 분석한다. 다양한 난이도와 알고리즘 유형에 대한 코딩 문제 해결력을 평가하고 분석하여, GPT 모델의 강점과 한계를 인간과 비교하여 파악한다.

GPT3.5와 4.0 모델은 낮은 난이도 문제에서는 인간보다 높은 정확도를 보이며, 중간 정도의 난이도 문제에서는 인간과 유사한 정확도를 보였다. 그러나 가장 어려운 문제에서는 인간이 20~30%의 문제 해결 능력을 보이는 반면, ChatGPT는 이러한 최상위 난이도의 문제를 해결하는데 실패하였다. 또한 GPT 모델은 단순 구현, 그래프 이론, 수학 유형의 문제에서 인간을 능가하는 정확도를 보였다. 그러나 다이내믹 프로그래밍, 그리디 알고리즘과 같은 복잡한 유형의 문제에서는 인간에 비해 다소 낮은 성능을 나타냈다. 참가자들은 다양한 알고리즘 분야의 코딩 문제를 효과적으로 해결할 수 있었으며, 이에 반해 ChatGPT는 학습 데이터의 양에 영향을 받아서 분야에 따라 코딩 문제 해결 능력에 다소 큰 편차를 보인다는 결과가 도출되었다.

GPT3.5 및 4.0이 생성한 코드와 인간이 작성한 코드의 성능을 비교한 결과, GPT는 참가자보다 메모리를 적게 사용하고 실행 시간도 짧다. 특히 코드 길이 비교에서 GPT 3.5 코드가 가장 짧게 생성되었으며, 참가자, GPT 4.0 순으로 GPT 3.5의 성능이 우수하게 나타났다. 이는 GPT4.0이 코드 모듈화 및 성능 최적화를 더 높은 수준으로 적용했기 때문에 코드가 길어진 결과로, 코드의 전체 구조 관리 및 재사용 측면에서 이점이 있다고 판단된다. GPT3.5, GPT4.0 및 참가자가 작성한 코드의 내용을 분석한 결과 차이점이 확인되었다. GPT 4.0은 클래스 사용, 모듈화, 주석 포함, 그리고 컨테이너 사용 등 다양한 평가 기준에서 우수한 코드를 생성하는 반면, GPT3.5는 클래스 구현을 거의 사용하지 않았으며, 참가자 역시 클래스 구현 및 주석, 컨테이너 사용 비율이 낮은 경향을 보였다. 오류 유형 측면에서 보면, GPT는 주로 간단한 정답 오류를 생성하며 참가자에 비해 더 적은 런타임 오류를 보여주는 강점이 있다. 또한, 오류 유형을 알려주고 다시 질의하면 정답률이 향상되는 경향이 있다. 이는 ChatGPT가 오류 수정 측면에서 활용될 수 있는 가능성을 시사한다.

결론적으로, GPT 모델은 기본적인 코딩 문제 해결에 활용 가능하나, 고난이도 문제나 복잡한 알고리즘 유형의 문제 해결에는 아직까지 한계가 존재한다. 코딩 문제 해결에 어려움을 겪는 사용자들에게 GPT가 올바른 정답을 제공함으로써 사용자가 코딩 테스트 문제에 대한 해답 확인이나 자체 코드와의 비교 과정에서 교육적 가치를 제공할 것으로 판단된다. 이 연구는 한국어 코딩 평가의 맥락에서 ChatGPT 모델의 성능을 이해하고, 인간과의 비교 결과를 제공함으로써 관련 분야에 기여한다.

VII. 논 의

코딩 과제를 해결하는 데 ChatGPT를 사용하는 학생들의 수가 증가함에 따라 교육 분야에서는 ChatGPT 사용의 허용 여부에 관한 논쟁이 치열해지고 있다. 본 연구에서 ChatGPT를 활용한 코딩 과제의 예시와 그에 대한 평가 체계에 대한 사례를 제시함으로써 교육 환경에서 ChatGPT 사용에 대한 논의에 기여하고자 한다. 구체적인 사례를 통해 ChatGPT를 코딩 교육에 통합하는 데 따른 잠재적인 이점과 도전에 대한 이해를 높이고자 한다. 표 8에서 ChatGPT를 활용한 코딩 과제를 제시하고 표 9에서 ChatGPT 코딩 과제에 대한 평가 체계를 제안한다.

표 8. ChatGPT를 활용한 코딩 과제 예시
Table 8. Example of coding assignment utilizing ChatGPT

Item	Description
Array operation	Design a function using ChatGPT to determine the average value of elements in an array. Optimize the time complexity using ChatGPT.
Linked list manipulation	Implement a function using ChatGPT to remove duplicate elements from a linked list. Ensure the function handles all test cases using ChatGPT.
Stack and queue operations	Design a stack and implement push and pop operations using ChatGPT. Optimize the overall structure, including modularization, using ChatGPT.
Tree traversal	Use ChatGPT to implement a function that determines the depth of a given node in a tree. Compare the solution with a student's code using ChatGPT.
Graph algorithm	Implement a function using ChatGPT to find the shortest path between two nodes in a weighted graph. Optimize the code using ChatGPT.

표 9. ChatGPT 활용 코딩 과제에 대한 평가 체계
Table 9. Evaluation for coding assignment utilizing ChatGPT

Item	Description
Accuracy of ChatGPT utilization	Evaluate whether the solutions for each task are accurate and produce the expected results, regardless of whether ChatGPT was used or not.
Adherence to coding standards	Assess the extent to which ChatGPT has been used to enhance code readability while adhering to coding standards.
Utilization of ChatGPT	Evaluate how ChatGPT was utilized in each task by reviewing the submission document and code. Assess its appropriateness for the given tasks.
Task-specific requirements	Evaluate whether each task satisfies specific requirements, including handling edge cases and optimization algorithms.
Improvement of code structure with GPT utilization	Compare and analyze the code structure improvements made by the student independently and with the assistance of ChatGPT. Evaluate the overall structure enhancement.
GPT utilization for problem-solving skills	Assess the creativity of the student in utilizing ChatGPT for problem-solving in the given tasks and evaluate the suitability of prompt engineering.

감사의 글

이 논문은 2022년도 성신여자대학교 학술연구조성비 지원에 의하여 연구되었음.

참고문헌

[1] S. Fergus, M. Botha, and M. Ostovar, “Evaluating Academic Answers Generated Using ChatGPT,” *Journal of Chemical Education*, Vol. 100, No. 4, pp. 1672-1675, March 2023. https://doi.org/10.1021/acs.jchemed.3c00087

[2] F. Antaki, S. Touma, D. Milad, J. El-Khoury, and R. Duval, “Evaluating the Performance of ChatGPT in Ophthalmology: An Analysis of Its Successes and Shortcomings,” *Ophthalmology Science,* Vol. 3, No. 4, 100324, December 2023. https://doi.org/10.1016/j.xops.2023.100324

[3] M. Cascella, J. Montomoli, V. Bellini, and E. Bignami, “Evaluating the Feasibility of ChatGPT in Healthcare: An Analysis of Multiple Clinical and Research Scenarios,” *Journal of Medical Systems*, Vol. 47, No. 1, 33, December 2023. https://doi.org/10.1007/s10916-023-01925-4

[4] P. M. Newton and M. Xiromeriti, “ChatGPT Performance on MCQ Exams in Higher Education. A Pragmatic Scoping Review,” *A Pragmatic Scoping Review*, June 2023. https://doi.org/10.35542/osf.io/sytu3

[5] S. Teebagy, L. Colwell, E. Wood, A. Yaghy, and M. Faustina, “Improved Performance of ChatGPT-4 on the OKAP Exam: A Comparative Study with ChatGPT-3.5,” *MedRxiv,* April 2023. https://doi.org/10.1101/2023.04.03.23287957

[6] J. C. F. de Winter, “Can ChatGPT Pass High School Exams on English Language Comprehension?,” *International Journal of Artificial Intelligence in Education*, September 2023. https://doi.org/10.1007/s40593-023-00372-z

[7] R. Bhayana, S. Krishna, and R. R. Bleakney, “Performance of ChatGPT on a Radiology Board-style Examination: Insights into Current Strengths and Limitations,” *Radiology*, Vol. 307, No. 5, 230582, June 2023. https://doi.org/10.1148/radiol.230582

[8] S. Frieder, L. Pinchetti, A. Chevalier, R.-R. Griffiths, T. Salvatori, T. Lukasiewicz, ... and J. Berner, “Mathematical Capabilities of ChatGPT,” arXiv: 2301.13867, July 2023. https://doi.org/10.48550/arXiv.2301.13867

[9] T. H. Kung, M. Cheatham, A. Medenilla, C. Sillos, L. De Leon, C. Elepaño, ... and V. Tseng, “Performance of ChatGPT on USMLE: Potential for AI-assisted Medical Education Using Large Language Models,” *PLOS Digital Health*, Vol. 2, No. 2, e0000198, February 2023. https://doi.org/10.1371/journal.pdig.0000198

[10] C. K. Lo, “What Is the Impact of ChatGPT on Education? A Rapid Review of the Literature,” *Education Sciences*, Vol. 13, No. 4, 410, April 2023. https://doi.org/10.3390/educsci13040410

[11] S. Biswas, “Role of ChatGPT in Computer Programming,” *Mesopotamian Journal of Computer Science*, Vol. 2023, pp. 8-16, February 2023. https://doi.org/10.58496/MJCSC/2023/002

[12] N. M. S. Surameery and M. Y. Shakor, “Use Chat GPT to Solve Programming Bugs,” *International Journal of Information technology and Computer Engineering,* Vol. 3, No. 1, pp. 17-22, January 2023. https://doi.org/10.55529/ijitc.31.17.22

[13] A. Kashefi and T. Mukerji, “ChatGPT for Programming Numerical Methods,” *Journal of Machine Learning for Modeling and Computing*, Vol. 4, No. 2, pp. 1-74, 2023. https://doi.org/10.1615/JMachLearnModelComput.2023048492

[14] E. Chen, R. Huang, H.-S. Chen, Y.-H. Tseng, and L.-Y. Li, “GPTutor: A ChatGPT-Powered Programming Tool for Code Explanation,” in *Proceedings of the 24th International Conference on Artificial Intelligence in Education (AIED 2023)*, Tokyo, Japan, pp. 321-327, July 2023. https://doi.org/10.1007/978-3-031-36336-8_50

[15] H. Tian, W. Lu, T. O. Li, X. Tang, S.-C. Cheung, J. Klein, and T. F. Bissyandé, “Is ChatGPT the Ultimate Programming Assistant - How Far Is It?,” arXiv: 2304.11938, April 2023. https://doi.org/10.48550/arXiv.2304.11938

[16] P. A. Massey, C. Montgomery, and A. S. Zhang, “Comparison of ChatGPT-3.5, ChatGPT-4, and Orthopaedic Resident Performance on Orthopaedic Assessment Examinations,” *Journal of the American Academy of Orthopaedic Surgeons,* Vol. 31, No. 23, pp. 1173-1179, December 2023. https://doi.org/10.5435/JAAOS-D-23-00396

[17] A. Taloni, M. Borselli, V. Scarsi, C. Rossi, G. Coco, V. Scorcia, and G. Giannaccare, “Comparative Performance of Humans versus GPT-4.0 and GPT-3.5 in the Self-Assessment Program of American Academy of Ophthalmology,” *Scientific Reports*, Vol. 13, 18562, October 2023. https://doi.org/10.1038/s41598-023-45837-2

[18] K. M. Caramancion, “News Verifiers Showdown: A Comparative Performance Evaluation of ChatGPT 3.5, ChatGPT 4.0, Bing AI, and Bard in News Fact-Checking,”

arXiv: 2306.17176, June 2023. https://doi.org/10.48550/arXiv.2306.17176

[19] Startlink. Baekjoon Online Judge [Internet]. Available: https://www.acmicpc.net/.

[20] Comparative Performance of Human versus ChatGPT 3.5 & 4.0 in the Korean Coding Test. Making Data Public [Internet]. Available: https://galvanized-order-7f0.notion.site/Comparative-Performance-of-Human-versus-ChatGPT-3-5-4-0-in-the-Korean-Coding-Test-90c6529f2f8544d7bda365d04761380c?pvs=4.

[21] Selenium. The Selenium Browser Automation Project [Internet]. Available: https://www.selenium.dev/documentation/.

[22] Python Package Index. Beautifulsoup4 [Internet]. Available: https://pypi.org/project/beautifulsoup4/.

최수지(Suzy Choi)

2023년 : 서울여자대학교 소프트웨어융합학과 (공학사)

2023년 ~ 현 재: 성신여자대학교 대학원 미래융합기술공학과 석사과정

※관심분야 : 딥러닝, 생성형 AI(Generative AI)

변혜원(Hae-Won Byun)

2004년 : KAIST 대학원 (공학박사-컴퓨터그래픽스)

1992년 : KAIST 대학원 (공학석사)

1990년 : 연세대학교 전산과학과 (공학사)

2006년 ~ 현 재: 성신여자대학교 AI융합학부 교수

※관심분야 : 컴퓨터 그래픽스, 딥러닝, 생성형 AI 등