컴퓨터교육학회 논문지 2026년 제29권 제3호
https://doi.org/10.32431/kace.2026.29.3.010

GPT-5를 활용한 코드 읽기 과제 자동 채점 탐색: 코드 목적 설명 과제를 중심으로

Exploring GPT-5-Based Automatic Scoring for Code Reading Tasks: Focusing on EiPL Tasks

박민규† · 최현종††
Mingyu Park† · Hyunjong Choe††

요약

본 연구는 GPT-5에 루브릭을 적용한 코드 목적 설명 과제 자동 채점 방식을 제안하고 채점 성능을 다각도로 탐색하는 것을 목표로 한다. 이를 위해 2022 개정 교육과정의 고등학교 정보 과목에 적합한 코드 목적 설명 과제 3개를 개발하여 J고등학교 1학년 약 390명을 대상으로 형성평가를 실시하고, 5년 이상의 교육 및 연구 경력을 보유한 교사 2인의 채점 결과를 기준 점수(ground truth)로 활용했다. 제안한 방식을 기존 CGBG(Code Generation Based Grading) 이진 채점 모델과 비교했으며, 교사 채점과의 일치도를 검증하기 위해 Cohen's Kappa를 사용하였다. 분석 결과, 제안한 방식은 CGBG 대비 평균 0.25 이상 높은 Cohen's Kappa 값을 보였다. 또한 제안한 방식과 교사 채점 간의 신뢰도를 두 채점 방식의 선형적 패턴 유사성, 순위 일관성, 그리고 허용 오차 기반 채점 일치율의 관점으로 해석하기 위해 각각 Pearson 상관계수, Kendall's Tau-b 상관계수, 그리고 Tolerance Adjusted Accuracy(TAA)로 분석하였다. 분석 결과, 두 상관계수는 강한 상관관계를 보였으며 1점 이내 오차 허용 시 일치율(TAA)은 85.7%에 도달하였다.

주제어 자동 채점, GPT, 코드 읽기, 코드 목적 설명 과제, 프로그래밍 평가

ABSTRACT

This study aims to propose an automatic scoring method for Explain in Plain Language(EiPL) tasks using GPT-5 with rubric based grading and to explore the feasibility of automatic scoring from multiple perspectives. To this end, three EiPL tasks for the high school Informatics under the 2022 revised national curriculum were developed, and a formative assessment was conducted to approximately 390 first-grade students at J High School. The scoring results of two teachers with more than five years of teaching and research experience were used as the ground truth. The proposed method was compared with the existing CGBG(Code Generation Based Grading) binary scoring model, and Cohen's Kappa was used to examine its agreement with human raters. The analysis showed that the proposed method achieved an average Cohen's Kappa value 0.25 higher than CGBG. In addition, the reliability between the proposed method and teacher scoring was examined from the perspectives of linear pattern similarity, rank consistency, and tolerance-based scoring agreement using Pearson's correlation, Kendall's Tau-b correlation, and Tolerance Adjusted Accuracy (TAA), respectively. The results showed strong correlations for both correlation coefficients, and when a tolerance of one point was allowed, the agreement rate(TAA) reached 85.7%.

Keywords Automatic scoring, GPT, Code reading, Explain in Plain Language, Programming assessment

†정회원 한국교원대학교 컴퓨터교육과 석사과정
††종신회원 한국교원대학교 컴퓨터교육과 교수(교신저자)

논문투고 2025년 11월 10일
심사완료 2026년 01월 03일
게재확정 2026년 01월 08일
발행일자 2026년 03월 22일

1. 서론

프로그래밍은 단순한 코드 입력 작업이 아니다. 겉으로 보기엔 키보드를 활용한 코드 입력에 불과할 수 있지만, 실제로는 작성 중인 코드가 의도대로 작동하는지, 더 나은 방법은 없는지, 오류는 없는지를 끊임없이 점검한다. 검토 중에는 과거에 이해했던 프로그램들을 회상하고, 돌이켜보기도 하며, 이를 현재 상황에 맞추어 적용해본다. 따라서 프로그래밍이란 코드를 입력하는 행위처럼 보이지만, 이것이 제대로 이루어지기 위해서는 프로그램을 읽고 이해하는 능력이 필수적이다.

본 연구에서 다루는 코드 목적 설명 과제는 학생들의 코드 이해 수준을 확인하고, 그 이해가 코드 작성으로 이어질 수 있는지 판단하는 데 중요한 역할을 한다. 코드 목적 설명 과제는 제시된 코드를 읽고 그 목적을 자연어로 설명하는 코드 읽기 활동이다. Lister 외(2006)는 전문가와 초보 프로그래머를 비교하면서, 제시된 코드가 무엇을 하려는지 목적을 파악하는 능력이 코드를 작성하는 과정으로 이어지는 중간 기술이 될 수 있다고 제안했다[1]. 그의 제안 이후, 코드 목적 설명 과제는 프로그래밍 이해를 평가하는 핵심적인 문항으로 연구에서 활용되었으며, 코드 이해 활동들과 코드 작성 사이에 유의미한 위계성이 존재함을 반복적으로 확인했다[2-5]. Lopez 외(2008)는 학생들의 풀이 데이터를 활용해 단계적 회귀분석을 실시한 결과, 코드를 추적하고 설명하는 문항이 코드 작성 문항의 변동성을 46% 설명한다는 점을 보고하였다[4]. Lister 외(2009)는“초보 학습자는 코드 작성 능력이 충분히 향상되기 위해서는 먼저 코드 추적과 설명 능력이 코드 작성을 지탱할 만큼 충분히 강해져야 하며, 그 이후에야 반복적인 코드 작성 연습을 통해 실질적인 향상이 이루어진다고 본다.”고 강조했다[3].

그러나 코드 목적 설명 과제의 교육적 가치가 여러 연구에서 제시되고 있음에도 불구하고, 실제 교수·학습 현장에서의 활용은 더디게 확산되고 있다[6]. Fowler 외(2021)의 연구 참여자들은 코드 목적 설명 과제의 채점에 대해 “학급 규모가 커 채점하기 어렵다”거나 “학생 답안이 매우 비구조적이라 채점이 어렵다”는 점을 지적하였다[7]. 이는 코드 목적 설명 과제의 교육적 의의와 별개로, 채점에 소요되는 시간과 노력, 그리고 평가의 일관성을 확보하기 어려운 현실적 부담이 현장에서 코드 목적 설명 과제의 활용을 제약하는 요인으로 작용하고 있음을 보여준다.

이러한 문제를 해결하기 위해 신뢰성 있는 코드 목적 설명 자동채점 방식에 대한 연구가 중요한 대안으로 제기된다. 신뢰도 높은 자동채점 방식이 채점 보조 수단으로 활용될 경우, 교사는 수기 채점에 소요되는 시간과 노력을 줄일 수 있으며, 이는 결국 코드 목적 설명 활동을 교수·학습 맥락에서 보다 안정적으로 확산시키는 데 기여할 수 있다. 이를 위해 지금까지 로지스틱 회귀[6]나 베이지안 접근을 활용한 자동채점 모델[8]이 제안되었으며, Fowler 외(2021)는 약 87%에서 89% 수준의 정확도를[6], Chen 외(2022)는 동료 평가 기반 접근의 가능성을 보고하였다[8]. 그러나 이러한 모델들은 학습 데이터 구축과 모델 훈련이 필수적이라는 점에서, 실제 학교 현장에서 활용하기에는 여전히 데이터 수집과 유지 관리의 부담이라는 현실적 한계를 지닌다.

최근 자연어처리 기술의 발달로 등장한 대규모 언어모델(Large Language Model, LLM)은 이러한 한계를 보완할 수 있는 실천적 대안으로 주목받고 있다. LLM은 대규모의 데이터를 기반으로 사전 학습되어, 추가적인 모델 학습 과정 없이도 사용자가 제시하는 프롬프트에 따라 자연어를 이해하고 생성할 수 있다는 점에서 실용적 장점을 지닌다. 현재까지 코드 목적 설명 과제의 자동채점에 LLM을 적용한 연구는 제한적으로 보고되고 있으며[9], 다양한 접근 방식이 제안되고 그 성능이 비교 및 검증될 필요가 있다.

따라서 본 연구는 루브릭 기반 접근을 통해 LLM을 활용한 코드 목적 설명 자동채점 방식을 제안하고, 그 결과를 기존 방식인 CGBG(Code Generation Based Grading) [9]과 비교한다. 또한 제안한 방식의 채점 신뢰도를 다각도로 검토한다. 이를 통해 궁극적으로 교수·학습 현장에서 코드 목적 설명을 활용한 코드 읽기 교육을 활성화하는 데 기여하고자 했다.

본 연구의 연구 질문은 다음과 같다.

1. 기존 채점 모델인 CGBG와 비교하였을 때, 제안한 채점 방식의 성능은 어떤 차이를 보이는가?
2. 제안한 채점 방식은 교사와의 점수 유사도 및 순위 일관성에서 어떠한 양상을 보이는가?

2. 이론적 배경

2.1 코드 읽기와 코드 추적

본 절에서는 프로그램 이해의 대표적인 활동인 코드 읽기와 코드 추적을 명확히 정의하고자 한다.

프로그래밍 교육에서 프로그램 이해는 핵심 연구 주제로 오랜 기간 다루어져 왔다. Soloway[10]는 프로그램 작성 과정에서 코드의 기능과 구조를 머릿속에서 시뮬레이션하며 프로그램 이해 활동을 수행한다고 보았다. 특히 초보 프로그래머는 프로그래밍 개념 학습을 위해 완성된 프로그램을 읽는 환경이 잦아 프로그램 이해 중심의 교수 · 학습이 중요하다[11].

초보 프로그래머가 프로그램 이해 능력을 향상하는 대표적인 학습 활동은 코드 읽기다. 코드 읽기는 완성된 프로그램을 분석하고 기능을 추론하는 과정이며, 일부 연구는 이를 코드 읽기와 코드 추적으로 구분한다[3-5, 12].

신수범[12]은 코드 추적을 디버깅의 관점에서 해석하여 실행 흐름을 따라가는 활동으로 제시하였다. 예를 들어, 완성된 프로그램의 예측 결과를 파악하는 활동, 디버깅 상황에서의 코드 수정이 해당된다. 반면, 코드 읽기 활동은 프로그램 목적이나 작성 이유를 파악하는 활동으로 제시했다. 본 연구에서도 신수범의 관점[12]을 차용하여 초보자의 프

로그램 이해 과정을 코드 읽기와 코드 추적으로 나누어 보다 구체적으로 논하고자 하였다.

2.2 코드 목적 설명 과제

본 절에서는 코드 목적 설명 과제의 뜻을 살펴보고, 코드 목적 설명 과제의 출제와 채점에서 연계되는 코드 목적 설명의 특징에 대해 정리하고자 한다.

코드 목적 설명 과제는 주어진 코드의 기능 또는 역할을 자연어로 설명하는 코드 읽기 활동이다. 이 과제는 프로그램의 작성 목적을 간결하고 명확하게 표현하는 능력을 평가한다.

```
a = 1
b = 2

## Explain what the highlighted section of the code does in Eng
lish or Korean.
## Do not describe it line by line.

tmp = a
a = b
b = tmp
```

Figure 1. An Example of EiPL Task

예를 들어, Fig. 1과 같은 코드에 대해 “색칠된 영역에 있는 코드는 어떤 역할을 하는가?”라는 질문이 제시된다면, 학습자는“두 변수의 값을 서로 바꾸는 코드이다”라고 서술할 수 있다.

이와 같이 코드 목적 설명 과제는 학생이 답안을 구성할 때 프로그래밍 언어가 아닌 자연어로 서술하도록 요구하는 평가 형태이다. 해당 문항은 EiPE(Explain in Plain English)의 이름으로 처음 제안되었으나, 영어권 이외의 다양한 언어권 학습자에게의 적용 가능성에 따라 EiPL(Explain in Plain Language)라는 명칭이 제안되었다[13]. 본 연구에서도 영어가 모국어가 아닌 학습자가 문제를 해결하는 교육적 맥락을 고려하여 영문으로는 ‘EiPL’, 한국어로는 ‘코드 목적 설명’을 사용한다. 따라서 EiPL 과제(문항)는 코드 목적 설명 과제(문항)로 표현한다.

코드 목적 설명 과제의 특징은 다음과 같다. 첫째, 해당 과제를 해결하는 과정에서 소스 코드에 포함된 변수의 역할을 파악해야 한다. 코드 목적 설명 과제는 단일 입력값의 실행 결과를 예측하는 것이 아니라, 다양한 입력 가능성을 전제하고 프로그램의 일반화된 목적을 기술하도록 요구한다. 따라서 코드 목적 설명 과제에서 각 변수는 특정 값에 대한 자리지기(placeholder)가 아니라, 알고리즘 전반에 걸쳐 유지되는 기능적 역할로 인식되어야 한다. 초보 프로그래머가 접하는 변수의 90% 이상은 ‘임시값 저장’, ‘최신값 보관’ 등 9가지 역할로 설명될 수 있는데[14], 변수 이름이 변수의 역할을 얼마나 암시하는지에 따라 코드 목적 설명 과제의 난이도 또한 달라진다[7].

예를 들어 Fig. 1의 변수 교환 코드에서 ‘tmp’ 변수는 temporary의 약어로, 임시 값 저장이라는 역할을 드러낸다. 반면 이를 ‘c’와 같이 의미가 드러나지 않는 이름으로 대체할 경우, 학생들은 해당 변수의 역할을 파악하는 데 더 큰 어려움을 겪는다. 따라서 문항 출제자는 학생이 변수 이름만으로 역할을 추측하는 것을 방지하려면 ‘c’와 같은 이름을 사용하는 것이 적절하며, 반대로 학습을 지원하는 비계(scaffolding)를 제공하고자 할 경우 ‘tmp’를 활용할 수도 있다. 이처럼 코드 목적 설명 과제에서는 출제 의도에 따라 변수명을 세심하게 설계할 필요가 있으며[7], 본 연구에서도 출제 목표에 따라 변수 이름을 조정하였다.

둘째, 학생이 프로그램의 전체적 의미를 이해했는지를 평가할 수 있다[7]. 즉, 코드의 실행 절차를 나열하기보다 프로그램의 목적을 중심으로 기술할 수 있는지를 본다. 예를 들어, Fig 1의 변수 교환 코드에 대해 초보자는 “tmp에는 a 값을, a에는 b 값을 넣고, b에는 tmp 값을 넣는다”고 기술하지만, 전문 프로그래머는 “두 변수의 값을 교환하는 코드”라고 설명한다[1]. 즉, 전문가는 줄별 실행 결과에 따른 갱신(코드 추적)이 아니라, 프로그램 요소들을 유기적으로 엮어 프로그램 목적을 구성(코드 이해)할 수 있다. 초보자가 프로그램 전체를 조망하지 못하는 현상은 ‘나무를 담고 있는 숲을 바라보지 못하는 현상’으로 비유되며, 코드 목적 설명과제에 대한 응답은 SOLO(Structure of the Observed Learning Outcomes) 분류 체계[15]를 통해 추상화 능력을 구별할 것이 제안되었다[1]. 3주의 프로그래밍 수업 이후 Fig. 1과 동일한 3줄짜리 코드 목적 설명 과제에서 절반의 학생이 오답을 보였다는 연구 결과는[2], 줄 단위의 실행을 넘어 전체적 관점에서 코드를 조망하는 훈련이 필요함을 시사한다.

본 연구에서는 코드 목적 설명 과제의 출제 의도가 답안의 추상성과 긴밀히 연관된다는 점에 주목하여, 추상성을 주요 채점 기준으로 반영한 Chen 외(2022)의 루브릭[16]을 활용하여 채점하였다. 구체적으로, 교사의 평가 과정에서 해당 루브릭을 적용하였을 뿐만 아니라, 본 연구에서 제안한 채점 프롬프트에도 이를 반영하여 답안의 추상성이 채점의 핵심 지표로 기능하도록 하였다.

2.3 GPT를 활용한 서술형 문항 자동 채점 연구

본 절에서는 GPT 기반 서술형 문항 자동 채점 연구의 동향을 살펴보고, 이를 토대로 본 연구의 방향과 프롬프트 전략을 정리하고자 한다.

최근 LLM의 발전은 서술형 평가 자동 채점 연구에 새로운 가능성을 제시한다. RAG(Retrieval-Augmented Generation) 기반 접근이나 파인튜닝(fine-tuning) 방식은 대규모 데이터와 연산 자원을 필요로 하지만, 교육 현장에서는 이를 구현하기 어려운 한계가 있다. 이에 프롬프트 엔지니어링(Prompt Engineering)을 기반으로 한 접근이 실용적 대안으로 주목받고 있다.

GPT의 문맥 이해력과 언어 생성 능력이 비약적으로 향상되면서[17], 물리[18], 세계지리[19], 수학[20] 등 다양한 교과 영역에서 프롬프트 기반 서술형 문항 자동 채점에 대

한 연구가 활발히 이루어지고 있다. 민태호와 이봉우(2024)는 채점 기준, 예시, 출력 요소의 조합에 따라 12가지 유형의 프롬프트를 설계하여 GPT의 채점 결과를 분석한 결과, 전문가보다 다소 관대한 채점 경향을 보였으나 Cohen's Kappa(0.12 ~ 0.34)와 Pearson 상관계수(0.41 ~ 0.67)를 통해 자동 채점의 가능성을 확인하였다[18]. 성정원과 신병철(2023) 은 프롬프트를 반영한 세 가지 유형의 채점 프롬프트를 실행한 뒤, 교사와의 Pearson 상관계수를 비교함으로써 GPT가 보조 평가자로서 잠재력을 지님을 제시하였다[19]. 또한 신병철 외(2024)는 순열과 조합 단원에 대해 다양한 프롬프팅 전략을 적용하여 GPT-4 기반 자동 채점을 수행하였으며, 0.8 수준의 상관관계를 확인하였다[20].

이와 같이 GPT 기반 서술형 자동 채점 연구들은 대체로 교사와의 유사성을 신뢰도 검증의 핵심 기준으로 삼고 있으며, 채점 기준을 LLM에게 반영하기 위한 프롬프트 설계가 주요 연구 방향으로 자리 잡고 있다. 본 연구 또한 교사의 평가를 검증을 위한 기준 점수로 활용하였으며, 채점 루브릭을 정교화한 프롬프트 설계에 초점을 두었다.

반면, GPT의 채점 일관성을 높이기 위해 활용할 수 있는 전략으로 Self Consistency(SC)와 Chain of Thought(CoT)가 있다. SC는 동일한 문제에 대해 여러 사고 경로를 탐색하고 가장 일관된 응답을 선택함으로써[21], 다양한 채점 관점 속에서 가장 일관된 품질의 결과를 출력할 수 있다. 반면, CoT는 복잡한 추론이 요구되는 문제에서 중간 추론 과정을 프롬프트에 작성하여 LLM에게 사고의 흐름을 명시적으로 안내하는 방식으로[22], 교사 채점자의 채점 흐름을 구체적으로 지시하여 안정적으로 채점을 진행할 수 있다. 이러한 두 전략을 활용하여, 채점자의 다양한 관점을 GPT에 반영하고 서술형 평가의 채점 성능을 비교하는 연구가 있다[20, 23, 24]. 이에 본 연구에서는 SC 유형의 프롬프트와 CoT 유형 프롬프트를 각각 적용하여 채점 성능을 비교 및 분석하였다.

2.4 코드 목적 설명 과제의 평가와 CGBG 모델

본 절에서는 프롬프트와 교사의 채점에 활용되는 코드 목적 설명 과제의 채점 루브릭[16]을 살펴보고, 본 연구에서 제안한 방식과 성능을 비교할 CGBG 모델에 대하여 살펴본다.

코드 목적 설명 과제를 실제 교육 현장에 적용하기 위해서는 채점 신뢰도 확보가 필수적이다. 코드 목적 설명 과제는 자연어로 서술한 답안을 포함하며 자연어는 프로그래밍 언어에 비해 다양한 해석이 가능하므로[6], 일관된 평가를 위해서는 명확한 루브릭을 기반으로 한 평가가 필요하다.

기존 연구에서는 SOLO 분류법에 근거한 4단계 평정 척도를 주로 활용해 왔다[1]. 그러나 4단계의 평정은 답안의 질적 차이를 충분히 구분하기 어렵고, 피드백 제공에 필요한 세분화된 기준이 부족하다는 한계가 지적되었다[25]. 또한 실제 학습자의 답변은 네 가지 수준보다 더 다양한 양상을 보이는 것으로 보고되었다[16].

이에 Chen 외(2020)는 코드 목적 설명 과제의 학습 결과를 추상성(abstraction), 정확성(correctness), 명확성(unambiguity)의 세 가지 차원으로 분석하고, 이들 간의 위계와 정도를 반영한 7점 척도의 루브릭을 제안하였다[16]. 본 연구에서는 이 루브릭을 적용하여 교사의 채점을 수행하였다. 또한 프롬프트 설계 시 해당 루브릭의 세 차원에 대한 설명과 루브릭을 함께 제시함으로써, 교사 평가와의 일관성을 확보하고자 하였다.

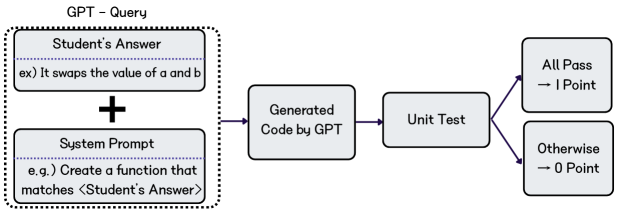

Figure 2. Pipeline of the CGBG Model [9]

한편, Smith와 Zilles(2024)는 기존 코드 목적 설명 자동 채점 모델[6, 8]이 대량의 학습 데이터를 요구한다는 한계를 극복하기 위해 CGBG 방식을 제안하였다[9]. 이 모델은 LLM의 뛰어난 코드 생성[26, 27]을 활용하는 방식으로, Fig. 2와 같이 학생의 자연어 설명과 코드 생성 프롬프트를 LLM의 입력으로 받아 코드를 생성하도록 한 뒤, 생성된 코드가 주어진 유닛 테스트(unit test)를 통과하는지를 기준으로 채점을 수행한다. 다시 말해, 학생 답변의 적절성을 LLM

Table 1. Research Procedure

Phase	Description
1. Task Design	EiPL (Explain in Plain Language) tasks were developed based on the 2022 Revised Curriculum for the high school subject Informatics.
2. Expert Review	Content validity and reliability of the developed tasks were verified through expert review.
3. Diagnositc Assessment	The tasks were administered as a diagnostic assessment during the first class of the Data Science class in the second semester of 2025, involving approximately 390 first-year students at J High School.
4. Human Rater Training and Calibration	Human Raters participated in training sessions to ensure shared understanding of the EiPL Task rubric and conducted a pilot scoring using 30 validation responses per question item, followed by multiple calibration meetings to finalize the scoring criteria.
5. Human Scoring	Human graded all student responses based on the finalized rubric.
6. Prompt Development	The GPT-based automatic scoring prompt was refined using the same 30 validation responses per question item as reference data.
7. Automatic Scoring and Comparison	Automatic scoring performed. Results were compared among human raters, the CGBG model, and the proposed GPT-based method.

이 생성한 코드의 실행 결과를 통해 간접적으로 검증하는 방식이다. 해당 방식을 제시한 연구[9]에 따르면 GPT-4를 기준으로 교사 채점자와의 Cohen's Kappa 가 평균 0.58 수준으로 나타났다.

CGBG 방식은 학습용 데이터 없이도 코드 목적 설명 과제를 자동 채점할 수 있다는 점에서 의의가 있다. 그러나 이 방식은 통과(1점) 또는 실패(0점)로만 구분되는 이진(binary) 평가 체계를 기반으로 하기 때문에, 학생 답안의 다양한 질적 수준을 충분히 반영하기 어렵다는 한계를 지닌다. 또한 코드의 줄 단위 기능을 나열하는 형태의 답안은 SOLO 분류법 상 낮은 수준으로 평가되지만, CGBG에서는 오히려 명시적인 프롬프트로 작동하여 채점 결과에 영향을 줄 수 있다. 이러한 불일치는 추상화를 평가의 핵심 목표로 하는 코드 목적 설명 과제의 타당도를 저해할 가능성이 있다. 마지막으로, CGBG 방식은 다른 자동 채점 방식과의 비교가 충분히 이루어지지 않았다는 점에서 추가적인 검토가 요구된다.

3. 연구 방법

3.1 연구 절차

본 연구의 절차는 Table 1과 같다. 2022 개정 교육과정의 고등학교 정보 과목 성취기준과 연계하여 코드 목적 설명 문항을 설계하고, 문항의 내용 타당도와 신뢰도 확보를 위해 전문가 검토를 실시하였다. 개발된 문항은 2025학년도 2학기에 진행된 '데이터 과학' 1차시 수업에서 파이썬 프로그래밍 능력을 점검하는 진단평가 형태로 활용되었다. 평가 대상은 J고등학교에서 정보 과목을 이수한 약 390명의 1학년 학생이며, 모두 1학기 동안 고등학교 정보 과목을 수강했다. 학생 답안은 비식별화된 상태로 수집되었으며, GPT 활용 전 응답을 검토하여 개인정보나 민감 정보가 포함되지 않도록 확인한 후 활용하였다.

학생들의 응답은 컴퓨터교육 분야 연구 경력과 5년 이상의 교육 경험을 모두 갖춘 두 명의 정보 교사가 채점자 합의와 채점 과정을 통해 수행되었다. 9월 2일에는 채점자 교육을 통해 코드 목적 설명 문항과 루브릭에 대한 이해를 공유하고, 무작위로 추출된 30개의 검증 데이터를 활용하여 가채점을 실시하였다. 이후 9월 8일과 15일 두 차례의 채점자 합의 과정을 거쳐 최종 채점 기준에 합의하였다. 채점 기준이 정립된 이후, 나머지 문항에 대해 채점을 진행했다.

자동 채점 프롬프트는 동일한 30개의 검증 데이터를 기반으로 점진적으로 개선하여 설계하였다. 각 문항별로 검증 데이터 30개를 제외한 나머지 학생 답안은 채점 성능 분석에 활용되었다.

최종적으로 CGBG 모델, 본 연구에서 제안한 채점 방식, 그리고 교사의 채점 결과를 비교·분석하였다.

3.2 코드 목적 설명 문항 출제

본 연구의 코드 목적 설명 문항 개발 절차는 성취기준 검토 및 적용 범위 결정, 문항 설계 원칙에 기반한 초안 개발, 전문가 타당도 검토 및 문항 수정, 그리고 현장 교사의 교수 타당도 검토로 이루어졌다.

먼저 2022 개정 교육과정의 고등학교 정보 과목에서 프로그래밍과 직접적으로 연관되는 성취기준을 검토하여 문항 개발에 적용 가능한 기준을 도출하였다. 이 과정에서 자료형에 관한 성취기준([12정03-04])은 개별 줄 단위의 이해에 초점을 두고 있어 코드 목적 설명 문항의 평가 목적과 직접적인 연관성이 낮다고 판단하여 제외하였다. 반면, 제어 구조 관련 성취기준([12정03-07])은 프로그램의 실행 흐름을 해석하고 의미를 설명하는 사고가 요구된다는 점에서 코드 목적 설명 과제에 적절한 문형이라고 판단하여, 다른 성취기준보다 상대적으로 더 많이 활용되었다. 최종적으로 [12정03-05]부터 [12정03-08]까지의 성취기준이 선택되었으며, 그 중 [12정03-07]이 중점적으로 활용되었다.

성취기준 선정이 완료된 후, 코드 목적 설명 문항의 특징을 토대로 문항 설계 원칙을 설정하였다. 코드 목적 설명 문항은 소스 코드를 통해 전체적인 의미를 구성할 수 있어야 한다. 이 때, 답안이 낯설거나 매우 특수한 상황에서만 통하는 맥락이라면 학생이 답안을 구성하는데 어려움이 있을 수 있다. 따라서 정보 교과서 8종을 분석하여 학습자가 실제 학습 중 접했을 것으로 예상되는 소재나 알고리즘을 중심으로 소스 코드를 구성하였다.

또한, 변수의 역할을 암시할 수 있는 변수 명 사용을 조절하여 불필요한 단서가 제공되지 않도록 유의했다. 구체적으로 알고리즘의 입력에 해당되는 변수는 입력값의 자료형과 의미를 알 수 있는 변수명(arr, num 등)을 활용하였으며, 알고리즘 내부에서 제시되는 변수의 경우 의미를 연결 짓기 힘든 변수명(a, i 등)을 활용하였다. 최종적으로 6개의 문항을 초기 문항으로 출제하였다.

Table 2. Composition of the Experts for Content Validity Review

Related Major and Degree	Position	Teaching / Research Experience
Ph.D. Candidate in Computer Education	Secondary School Teacher	5 – 10 years
Ph.D. Candidate in Computer Education	Secondary School Teacher	Less than 5 years
Ph.D. Candidate in Computer Education	Secondary School Teacher	Less than 5 years
Master Degree in Computer Education	Secondary School Teacher	10 – 15 years
Master Degree Candidate in Computer Education	Secondary School Teacher	10 – 15 years
Master Degree Candidate in Computer Education	Secondary School Teacher	5 – 10 years
Master Degree Candidate in Computer Education	Researcher	Less than 5 years
Master Degree Candidate in Computer Education	Researcher	Less than 5 years
Master Degree Candidate in Computer Education	Researcher	Less than 5 years
Master Degree Candidate in Computer Education	Researcher	Less than 5 years

개발된 문항은 Table 2의 전문가를 대상으로 한 두 차례의 전문가 타당도 검토를 통해 내용 타당도를 검토하였다. 검토 자료에는 성취 기준, 출제 근거, 소스 코드와 발문, 예상 답안, Chen 외(2020)의 루브릭[16]을 포함하였으며, 평가는 5점 리커트 척도(성취기준 반영 여부, 문항 표현의 명확성, 예상 답안의 타당성, 문항 난이도의 적절성)와 자유서술형 의견으로 구성되었다. 그 결과 모든 문항이 CVI(Content Validity Index) 기준값인 0.8 이상이었으나, 성취기준과 문항의 평가 목표가 다소 일치하지 않거나, 예시 답안의 표현과 관련하여 세부 조정이 필요하다는 서술형 의견이 제시되었다. 이에 따라 평가 목표가 부각되도록 문항을 수정하고, 예시 답안의 표현을 명확히 하였다.

마지막으로 현장 교사와의 논의를 통해 실제 교수 · 학습 맥락에서의 적용 가능성을 검토했다. 검토 결과 학습자의 학습 경험이 주로 제어 구조([12정03-07]) 중심으로 이루어져 있다는 점이 확인되었으며, 이에 따라 해당 성취기준과 연계된 문항을 최종적으로 선정하였다. 또한 학습자가 익숙한 표현으로 문항을 수정하였다. 최종 확정된 문항은 Table 3과 같다.

Table 3. EiPL Task Items

Question Prompt	Describe clearly and concisely the purpose (or role) of the highlighted section in the program.	
Item No.		Sample Answer
Item 1	`arr = [1, 3, 2, 4]` `a = arr[0]` `for i in range(len(arr)):` `  if arr[i] < a:` `  a = arr[i]` `print(a)`	It finds the minimum value in the list arr.
Item 2	`num = 10` `check = True` `if num < 2:` `  check = False` `for i in range(2, num):` `  if num % i == 0:` `  check = False`	It checks whether num is a prime number.
Item 3	`arr = [1, 3, 4, 7, 8]` `check = True` `for i in range(len(arr)-1):` `  if arr[i] > arr[i+1]:` `  check = False` `print(check)`	It checks whether the given array is sorted in ascending order.

3.3 프롬프트 설계

GPT-5가 채점을 수행하기 위한 최소한의 맥락으로 발문(소스 코드 포함), 채점 기준, 학생 답안, 역할과 목적을 프롬프트에 반영하였다. Chen 외(2020)의 루브릭[16]과 제작 근거가 되는 세 가지 정성적 차원(정확성, 추상성, 명확성)을 기재하여 루브릭에 대한 이해를 높이고자 하였다. 또한 채점 결과와 이에 대한 근거를 JSON 형태로 출력하도록 규칙(Output Format)을 명시하였다. 답안의 총체적 인상이 평가에 반영된다는 점을 고려하여, 교사 채점자의 채점 예시와 그 근거 6개를 프롬프트에 포함하였다. 이를 통해 GPT-5가 교사 채점자의 판단 기준과 근거를 학습하도록 하였다.

일관된 평가 수행을 위해 두 가지 유형의 프롬프트를 제작하였다. 첫째, CoT기반 프롬프트(P1)에서는 교사가 채점자 합의 과정에서 보였던 절차인 '답안 및 문제 이해 → 핵심 아이디어 추출 → 차원별 평정 → 루브릭 기반 점수 산출 → 채점 예시와의 비교 → 채점 예시표 재해석 및 재채점' 을 지시사항(Instructions)에 단계적으로 명시하였다. 둘째, SC기반 프롬프트(P2)에서는 GPT-5가 세 개 이상의 사고 경로를 생성하고 그중 가장 일관된 판단 결과를 출력하도록 지시사항(Instructions)에 지정하였다.

Table 4. Description of Prompt Component

Component	Role
Role and Objective	Establishes evaluator identity and purpose
Instructions	Guides consistent scoring with CoT strategy
Evaluation Dimensions	Provides three dimensions that constitute rubric
Detailed Rubric	Provides detailed scoring standards
Output Format	Enables structured output
Question	Provides contextual grounding for evaluation
Scoring Examples	Supplies reference for consistent scoring
Student Answer	Serves as the target of evaluation

각 문항별로 수집된 데이터 중 문항 당 30개를 할애하여 프롬프트 설계를 위한 검증 데이터로 활용하였다. 특히 30개 문항에 대한 채점 성능을 확인하여 P1과 P2 중 더 높은 성능의 프롬프트를 최종 선택하고자 했다. 교사와 P1, 교사와 P2의 점수 차이에 대해 Wilcoxon Signed Rank Test를 수행한 결과 두 프롬프트 채점 결과의 평균이 동일하다는 귀무가설을 기각할 수 없었다(p=0.4795). 그러나 CoT는 GPT의 사고 과정을 보다 명시적으로 안내하고[22], 루브릭 및 문항 맥락과 결합될 경우 추론 과정이 채점 목적에 맞게 정렬되어 자동 채점의 안정성을 높이는 요인으로 보고된 바 있다[24]. 이러한 선행 연구와 본 연구의 검증 결과를 종합하여, 교사 채점과의 차이가 상대적으로 작은 P1을 최종 프롬프트로 선정하였다. 최종 프롬프트의 구성 요소는 Table 4와 같으며, 자세한 프롬프트는 부록의 Table 10에 제시하였다.

모든 GPT 실행 과정은 GPT-5(gpt-5-2025-08-07) 모델의 API 호출로 Python에서 실행되었으며, 클라이언트 단위별로 응답을 분리하여 이전 응답이 현재에 영향이 없도록 하였다. Reasoning Effort와 같은 매개변수는 모델 초기값으로 고정하여 사용했다.

3.4 측정 지표

본 연구에서는 채점 성능을 다각도로 검증하기 위해, 자동 평가 연구에서 일반적으로 활용되는 Pearson 상관계수 외에도 Kendall's Tau-b와 Tolerance Adjusted Accuracy(TAA) 값을 함께 산출하였다. Chang과 Ginter(2024)는 평가자에 따라 점수의 순위가 변동되는 현상이 학생들에게 평가의 공정성 훼손으로 인식될 수 있음을 지적하였다[28]. 동일한 맥락에서 본 연구는 평가의 공정성을 정량적으로 나타내기 위해 τ_b를 사용하였다.

Kendall's Tau(τ)는 두 변수 간의 순위 일치 정도를 비모수적으로 검정하는 지표이다. 자동 채점 결과를 x_i, 교사의 채점 결과를 $y_i(i=1,2,3,\cdots, n)$, C를 두 채점 결과의 순위가 일치하는 쌍의 수, D를 불일치하는 쌍의 수라 하면, τ는 다음과 같이 계산된다[29].

$$\tau = \binom{n}{2}^{-1}(C-D)$$

τ는 가능한 조합 중 순위가 일치하는 비율과 그렇지 않은 비율의 차로 이해할 수 있다. 따라서 τ를 활용하면 서로 다른 평가 방식에 따라 결과가 뒤집히는 정도를 정량화할 수 있다[30]. 본 연구에서는 동점인 경우를 보정한 Kendall's Tau-b (τ_b)를 사용하였으며, τ_b는 다음과 같이 계산한다[31].

$$\tau_b = \frac{C-D}{\sqrt{C+D+T_x}\sqrt{C+D+T_y}}$$

(단, T_x와 T_y는 각각 각각 자동 채점 결과와 교사 채점 결과에서 동점인 쌍의 수)

TAA는 교사는 자동채점 결과를 100% 신뢰하지는 않으며, 다만 인간 점수와의 차이가 주어진 범위인 ε 안에 있다면 채점 모델의 결과를 받아드릴 것이라는 현실적인 인식에서 제안되었다[28]. 자동채점 결과(x_i)와 교사의 채점 결과(y_i)에 대하여 ($_i=1,2,3,\cdots, n$), ε차이를 용인하는 TAA_ε는 다음과 같이 정의된다.

$$TAA_\epsilon = \frac{1}{n}\sum_{i=1}^{n} 1(|x_i - y_i| \le \epsilon) \times 100(\%)$$

ε값이 0인 경우는 정확도(Accuracy)와 동일하다.

4. 연구 결과

본 장에서는 교사 간의 채점 신뢰도를 확인하고, 기존 자동 채점 모델(CGBG)과 제안한 GPT 기반 채점 방식의 성능을 비교하였다. 또한 제안한 방식이 교사의 판단과 어느 정도 일관성을 유지하는지를 Pearson 상관계수, TAA, Kendall's Tau-b 지표를 통해 다각도로 검증하였다. 이를 통해 자동 채점의 신뢰도와 순위 일관성을 종합적으로 분석하였다.

4.1 교사의 채점 결과

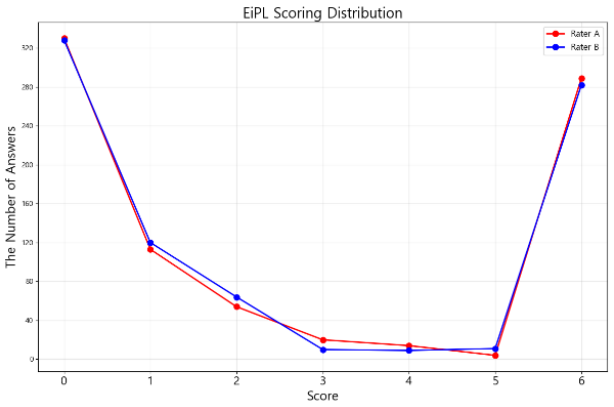

Figure 3. EiPL Scoring Distribution by Human Scorer

교사 채점 결과의 분포는 Fig. 3과 같다. 두 채점자 모두 0점과 1점 구간, 그리고 6점에서 높은 빈도를 보였으며, 2점부터 5점까지의 중간 점수 구간에서는 응답 수가 상대적으로 적었다. 이는 상당수의 응답이 정확성 또는 추상성이 매우 낮거나, 반대로 충분히 정확하고 추상적 설명 수준을 갖춘 답변이었다는 점을 의미하며, 부분적 이해 수준의 응답은 제한적으로 나타났음을 시사한다.

두 채점자의 분포는 Fig 3.과 같이 전체적으로 유사하게 나타났다. 채점자 간 상관계수를 산출한 결과, Pearson 상관계수는 0.9로 채점자 간 높은 양의 상관관계를 보였다. 채점자 간 신뢰도의 한 증거로 성태제(2002)는 상관계수가 0.6 이상을 제시하였다[32]. 따라서 교사 채점자 간 상관계수 값이 0.6 이상이라는 사실은 성태제[32]가 제시한 신뢰도 기준을 충족하였다.

그러나 상관계수만으로는 점수 변동의 원인이 무엇인지 설명하기 어렵기 때문에, 본 연구에서는 추가적으로 일반화가능도 분석을 실시하였다. 일반화가능도 분석은 관찰 점수의 분산을 세부 요인별로 분리하여 점수 변동이 어떤 요소에서 비롯되었는지를 규명하는 분석 방법으로, 오차의 크기뿐만 아니라 각 요인이 전체 관찰 점수 변동에 기여하는 정도를 함께 파악할 수 있다[32]. 본 연구에서는 채점자(Rater), 문항(Item), 그리고 피험자(Person)가 점수 변동에 미치는 정도를 파악한다.

Table 5. Results of the Generalizability Analysis

Factor	Variance (Proportion)
Person	2.9096 (0.3915)
Item	0.5255 (0.0707)
Rater	0.0001 (0.0000)
Person × Item	3.8100 (0.5126)
Person × Rater	0.0104 (0.0014)
Rater × Item	0.0004 (0.0001)
Person × Item × Rater	0.1767 (0.0238)
Total	7.4327 (1.0000)

일반화가능도 분석을 통해 분산 성분을 추정한 결과는 Table 5와 같다. 전체 요인 중 채점자(Rater)와 관련된 요인은 채점자(Rater), 채점자×문항(Rater×Item), 채점자×피험자(Rater×Person)요인이며, 이들이 차지하는 분산 비율은 각각 0.00%, 0.04%, 0.14%로 총 0.18%에 불과하였다. 반면, 분산의 대부분은 피험자×문항(Person×Item) 요인(51.26%)과 피험자(Person) 요인(39.15%)으로 설명되어 두 요인의 합이 전체 분산의 90% 이상을 설명하였다.

이는 점수 변동이 주로 학습자 개인의 수행 차이와 문항 특성(피험자×문항, 피험자)에 의해 발생하며, 채점자와 관련된 요인(채점자, 채점자×문항, 채점자×피험자)이 점수 변동에 기여하는 비중은 매우 낮음을 의미한다. 다시 말해, 특정 채점자가 특정 문항이나 특정 피험자 집단에 대해 일관되지 않게 채점했을 가능성은 상대적으로 낮으며, 채점 결과가 채점자에 의해 좌우되지 않고 비교적 안정적으로 유지되었음을 보여준다.

상관분석과 일반화가능도 분석 결과를 종합해보았을 때, 채점자 간 신뢰도를 확보했다고 판단했다.

4.2 CGBG 모델의 채점 결과와 제안한 방식의 채점 결과 비교

본 연구에서 비교 대상으로 설정한 CGBG 모델은 이진 채점 체계를 전제로 한다. Chen 외[16] 역시 CGBG 모델을 검증할 때 교사 채점 결과 추상성, 정확성, 명확성이 모두 높은 수준에 도달한 경우를 정답(1점)으로 분류하고, 그 외의 경우를 오답(0점)으로 처리하였다. 따라서 본 연구에서도 비교를 위해 동일한 기준으로 교사 채점 결과 만점에 도달한 경우에는 1점으로 처리하였고, 그렇지 못한 경우에는 0점으로 기술하였다. 이는 서로 다른 척도로 운영되는 두 채점 방식을 동일한 단위로 비교하기 위한 절차였다.

그럼에도 불구하고, 점수의 이진화(binarize)는 루브릭이 포착하고자 한 중간 수준의 정보가 축소되어 과소 대표될 가능성을 내포한다. 또한, 이진화 과정은 0점 범주에 응답이 상대적으로 많이 집중되는 분포를 형성할 수 있으며, 이는 주변분포(marginal distribution)의 편향에 민감하게 반응하는 Cohen's Kappa 해석에 영향을 줄 수 있다[33]. 따라서 본 연구에서 제시된 Cohen's Kappa 값은 이러한 분석 환경을 고려하여 신중하게 해석될 필요가 있다. 향후 연구에서는 이진 점수 체계를 전제로 한 다양한 LLM 기반 자동 채점 모델이 제시될 필요가 있으며, 이러한 모델을 CGBG와 동일한 기준에서 비교·검증하는 추가 연구가 요구된다.

기존 자동채점 모델(CGBG)과 본 연구에서 제안한 GPT 기반 채점 방식 간의 성능 차이를 비교하였다. 두 방식과 교사 간의 일치도를 Cohen's Kappa 계수(κ)를 통해 산출한 결과는 각각 Table 6과 같다.

Table 6. Scoring Agreement Using Cohen's Kappa

Type	Rater	Item	κ per item	κ for all Items	Average κ per Rater
CGBG vs Human	A	1	0.7169	0.6467	0.6361
		2	0.5773		
		3	0.5714		
	B	1	0.6932	0.6255	
		2	0.5252		
		3	0.5793		
The Proposed vs Human	A	1	0.9158	0.8837	0.8953
		2	0.8210		
		3	0.8688		
	B	1	0.9287	0.9070	
		2	0.8382		
		3	0.9158		

기존 CGBG 모델과 교사 간의 Cohen's Kappa(K_1)는 문항별로 0.5714부터 0.7169까지, 전체 문항 평균은 평가자별로 0.6467, 0.6255로 나타났다. 반면, 제안한 방식과 교사 간의 Cohen's Kappa(K_2)는 문항별로 0.8210부터 0.9287까지, 전체 문항 평균은 평가자별로 0.8837, 0.9070으로 나타났다. 모든 문항에서 제안한 방식의 일치도(K_2)가 기존 모델의 일치도(K_1)를 상회하였으며, 전체 문항 평균에서도 약 0.2 이상 상승했다.

4.3 제안한 방식의 채점 결과와 교사 평가의 결과 비교

본 연구는 제안한 방식의 채점 결과가 교사의 결과와의 일관성과 유사성을 다면적으로 검증하는 것을 목표했다. 이를 위해 첫째, Pearson 상관계수를 통해 교사와 제안한 방식 간의 선형 유사성을 분석하였고, 둘째, Tolerance Adjusted Accuracy(TAA)를 통해 오차 허용 범위 내에서의 일치도를 평가하였다. 마지막으로 Kendall's Tau-b를 통해 평가자 간 점수 순위의 일관성을 확인하였다.

Table 7. Pearson Correlation between the proposed Method and Human Raters

Item	Rater A	Rater B	Mean of Raters
Item 1	0.9318***	0.9280***	0.9331***
Item 2	0.8929***	0.8944***	0.9047***
Item 3	0.8846***	0.9023***	0.8990***
Mean of Items	0.9111***	0.9151***	0.9189***

(p* < 0.05, p** < 0.01, p*** < 0.001)

Pearson 상관분석 결과는 Table 7과 같다. 제안한 자동 채점 방식과 교사 간의 점수 일치도는 모든 문항에서 매우 높은 수준으로 나타났다. 문항별로 살펴보면, 제안한 방식과 평가자 A의 상관계수는 0.8846에서 0.9318 사이로 분포하였으며, 평가자 B의 경우 0.8944에서 0.9280 사이의 값을 보였다. 두 평가자의 평균 점수와의 상관계수 역시 0.8990에서 0.9331 사이로 나타나 전반적으로 매우 선형 관계가 유지되었으며, 모두 통계적으로 유의하였다($p < .001$). 0.6 이상의 상관계수를 신뢰도의 근거로 볼 때[32], 본 결과는 제안한 자동 채점 방식이 교사의 채점을 보조할 수 있는 잠재력을 지님을 시사한다.

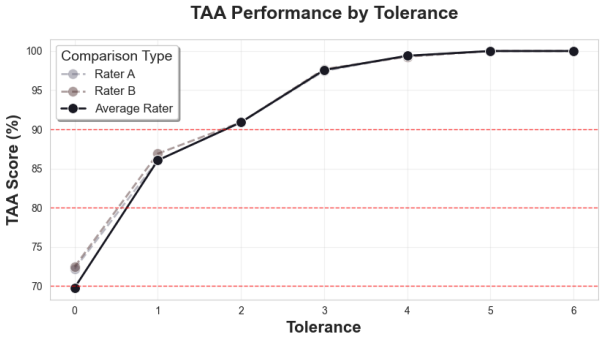

Figure 4. TAA Performance by Tolerance

허용 오차 범위에 따른 TAA 변화는 Fig 4.에 제시하였다. 제안한 자동 채점 결과와 교사의 평가의 점수가 완전히 일치해야 하는 조건(ε=0)에서는 TAA가 약 70% 수준으로 나타났다. 이후 허용 오차가 증가함에 따라 TAA는 전반적으로 상승하며 최댓값인 100%에 수렴하는 양상을 보인다.

Table 8. TAA Improvement by Tolerance(ε)

Tolerance(ε)	Improvement Rate(%p)	Rater A(%)	Rater B(%)
0	-	72.2	72.5
1	+15.9	85.7	86.6
2	+4.9	90.5	90.8
3	+6.9	97.5	97.7
4	+1.8	99.4	99.7
5	+0.6	100.0	100.0
6	+0.0	100.0	100.0

(p* < 0.05, p** < 0.01, p*** < 0.001)

Table 8은 ε에 따른 TAA의 변화를 더욱 구체적으로 설명한다. ε가 0에서 1로 증가할 때, TAA 값은 69.8%에서 15.9%p.만큼 증가하는 양상을 보인다. 이후 ε가 2 이상으로 증가하더라도 TAA 향상 폭은 점차 감소하였으며, 해당 지점 이후의 변화 경향은 비교적 완만했다. 이러한 패턴은 자동 채점 방식과 인간의 채점 결과가 차이가 나는 경우 그 오차는 1인 경우가 가장 많음을 시사한다. 또한 허용 오차가 2일 때 자동 채점 결과와 교사의 채점 결과 간의 차이가 일정 수준 수렴하기 시작하는 경향이 나타났음을 보여준다.

Table 9. Kendall's Tau – b Correlation between the proposed Method and Human Raters

Item	Rater A	Rater B	Mean of Raters
Item 1	0.8582***	0.8604***	0.8655***
Item 2	0.7978***	0.7983***	0.8012***
Item 3	0.7845***	0.8090***	0.7983***
Mean of Items	0.8200***	0.8277***	0.8264***

(p* < 0.05, p** < 0.01, p*** < 0.001)

Kendall's Tau-b(τ_b) 분석 결과는 Table 9와 같다. 제안한 자동 채점 방식과 교사 간의 순위 일치도는 전반적으로 높은 수준으로 나타났다. 문항별 τ_b값은 0.7845에서 0.8604 사이였으며, 모두 통계적으로 유의하였다($p < .001$). 이는 점수의 절대적인 값뿐만 아니라 순위 간의 일관성을 기준으로 하더라도 자동 채점 결과가 교사의 판단과 유사한 경향을 보임을 의미한다.

5. 결론 및 논의

본 연구는 GPT-5 기반의 코드 목적 설명 과제의 자동 채점 방식을 제안하고, 그 신뢰도를 검증하는 것을 목표로 하였다. 이를 위해 2022 개정 교육과정 고등학교 정보 과목의 성취기준을 토대로 코드 목적 설명 문항을 설계하고, 진단평가를 실시하여 교사 채점 결과와 자동 채점 결과를 비교 및 분석하였다.

기존 CGBG 모델과의 비교 결과, 제안한 방식은 모든 문항에서 약 0.2 이상 높은 Cohen's Kappa 값을 보였다. McHugh(2012)에 따르면 Cohen's Kappa 값에 따른 일치 수준은 대략 0.1 또는 0.2 단위로 구분되므로[34], 본 연구에서 확인된 Cohen's Kappa 값 개선은 채점 일치 수준이 한 단계 상승한 의미 있는 개선 효과로 해석될 수 있다.

또한 Pearson 상관계수와 Kendall's Tau-b 분석 결과, 제안한 방식은 교사 채점과 유의미한 강한 양의 상관 관계를 보였다. 이는 교사 채점과 자동 채점 간 점수의 선형적 관계뿐만 아니라, 학생 수행의 상대적 순위 판단 역시 안정적으로 보존되었다는 점을 의미한다. 더불어 TAA 분석에서 대부분의 오차가 1점 수준에서 가장 큰 향상 폭을 보인다는 점은, 제안한 자동 채점 방식을 전적으로 대체 수단으로 활용하기보다는 교사가 약 1점 수준의 오차 가능성을 고려하여

보조 채점 도구로 활용하는 경우 보다 높은 일치도를 보임을 시사한다.

이러한 결과를 토대로 한 논의는 다음과 같다.

첫째, 본 연구의 결과는 LLM 기반 자동 채점이 일정 수준의 평가 신뢰도를 확보할 수 있음을 보고한 선행연구들[19, 20]과 그 흐름을 같이 한다. 이는 GPT 기반 자동 채점이 수학, 과학 영역을 넘어 프로그램 이해에 관한 평가 영역에서도 적용 가능성이 있음을 실증적으로 제안한 결과라 할 수 있다.

둘째, 제안한 방식은 기존 CGBG 모델이 갖는 '이진 채점'의 한계를 넘어 보다 세밀한 수준의 채점이 가능함을 실증적으로 확인하였다.

셋째, CoT 기반 프롬프트 전략이 채점 과정의 추론 흐름을 구조화하고, 루브릭을 통해 GPT의 논증 맥락을 정렬하는 데 기여했음을 확인하였다. 이는 CoT 전략이 채점의 일관성과 안정성을 향상시킬 수 있다는 기존 연구[24]의 논의와도 일치하며, 프로그래밍 이해 과제에서도 동일한 효과가 재현될 수 있음을 시사한다.

본 연구의 한계는 다음과 같다.

첫째, 연구가 단일 학교, 단일 학년, 특정 단원을 중심으로 수행되었기 때문에 결과의 일반화 가능성에는 제한이 있다. 따라서 연구 대상을 넓혀 추가적인 검증이 필요하다.

둘째, 본 연구에서 수집된 학생 답안의 점수 분포가 특정 점수대에 상대적으로 집중되는 경향을 보였다. 이러한 분포 특성은 학습자의 점수가 양분되는 평가 맥락을 반영한 결과로 볼 수 있으나, 동시에 중간 수준의 다양한 응답이 제한적으로 포함되었다는 점을 의미한다. 이로 인해 자동 채점 모델은 빈도가 많은 특정 점수대에 더 적합해지고, 상대적으로 드물게 나타나는 중간 수준 응답에 대해서는 채점 기준을 안정적으로 일반화하지 못할 가능성이 존재한다. 따라서 다양한 점수 분포를 포함하는 데이터 셋에서 동일한 수준의 안정적 성능이 재현되는지에 대해 추가적인 검증이 필요하다.

그럼에도 불구하고 본 연구는 고등학생을 대상으로 한 코드 목적 설명 문항을 설계하고, GPT 기반 자동 채점이 일정 수준 이상의 타당성과 신뢰성을 확보할 수 있음을 실증적으로 확인하였다는 점에서 의의를 갖는다. 이러한 결과는 교수·학습 현장에서 코드 목적 설명 과제 도입을 통한 코드 읽기 교육 활성화를 지원하여, 코드 읽기에 대한 실천적 논의가 후속 연구와 평가로 이어질 수 있는 기초를 제공한다.

참고문헌

[1] Lister, R., Simon, B., Thompson, E., Whalley, J. & Prasad, C. (2006). Not seeing the forest for the trees: Novice programmers and the SOLO taxonomy. *ACM SIGCSE Bulletin, 38*(3), 118–122. https://doi.org/10.1145/1140123.1140157

[2] Corney, M., Lister, R., & Teague, D. (2011). Early relational reasoning and the novice programmer: Swapping as the'hello world'of relational reasoning. *Proceedings of the Thirteenth Australiasian Computing Education Conference*, Perth, Australia, 95–104.

[3] Lister, R., Fidge, C., & Teague, D. (2009). Further evidence of a relationship between explaining, tracing and writing skills in introductory programming. *ACM SIGCSE Bulletin, 41*(3), 161–165. https://doi.org/10.1145/1595496.1562930

[4] Lopez, M., Whalley, J., Robbins, P., & Lister, R. (2008). Relationships between reading, tracing and writing skills in introductory programming. *Proceedings of the Fourth International Workshop on Computing Education Research*, Sydney, Australia, 101–112. https://doi.org/10.1145/1404520.1404531

[5] Venables, A., Tan, G., & Lister, R. (2009). A closer look at tracing, explaining and code writing skills in the novice programmer. *Proceedings of the Fifth International Workshop on Computing Education Research Workshop*, Berkeley, USA, 117–128. https://doi.org/10.1145/1584322.1584336

[6] Fowler, M., Chen, B., Azad, S., West, M., & Zilles, C. (2021). Autograding "Explain in Plain English" questions using NLP. P*roceedings of the 52nd ACM Technical Symposium on Computer Science Education*, 1163–1169. NewYork, USA, https://doi.org/10.1145/3408877.3432539

[7] Fowler, M., Chen, B., & Zilles, C. (2021). How should we 'Explain in plain English'? Voices from the Community. *Proceedings of the 17th ACM conference on international computing education research*, Online, 69–80. https://doi.org/10.1145/3446871.3469738

[8] Chen, B., West, M., & Zilles, C. (2022). Peer-grading "Explain in Plain English": A Bayesian Calibration Method for Categorical Answers. P*roceedings of the 53rd ACM Technical Symposium on Computer Science Education*, Providece, USA, 133–139. https://doi.org/10.1145/3478431.3499409

[9] Smith, D., & Zilles, C. (2024). Code Generation Based Grading: Evaluating an Auto-grading Mechanism for "Explain-in-Plain-English" Questions. *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1*, New York, USA, 171–177. https://doi.org/10.1145/3649217.3653582

[10] Soloway, E. (1986). Learning to program= learning to construct mechanisms and explanations. *Communications of the ACM, 29(9)*, 850–858. https://doi.org/10.1145/6592.6594

[11] Schulte, C. (2008). Block Model: An educational model of program comprehension as a tool for a scholarly approach to teaching. *Proceedings of the fourth international workshop on computing education research*, NewYork, USA, 149–160. https://doi.org/10.1145/1404520.1404535

[12] Shin, S. (2017). Research on the Assessment Criteria of Programming Education based on Bloom's Taxonomy in the Elementary and Secondary School. *Journal of The Korean Association of Information Education, 21*, 547–555. https://doi.org/10.14352/jkaie.2017.21.5.547

[13] Smith IV, D., Kumar, V., & Denny, P. (2024). Explain in plain language questions with Indic languages: Drawbacks, affordances, and opportunities. *17th Annual*

ACM India Compute Conference, Gandhinagar, India, 3–17. https://doi.org/10.1007/978-3-031-84391-4_1

[14] Sajaniemi, J. (2002). An empirical analysis of roles of variables in novice-level procedural programs. *Proceedings IEEE 2002 Symposia on Human Centric Computing Languages and Environments*, Arlington, USA, 37–39.

[15] Biggs, J., & Collis, K. (1981). *Evaluating the quality of learning: The SOLO taxonomy (structure of the observed learning outcome)*. Elsevier Science & Technology. https://doi.org/10.1016/C2013-0-10375-3.

[16] Chen, B., Azad, S., Haldar, R., West, M., & Zilles, C. (2020). A Validated Scoring Rubric for Explain-in-Plain-English Questions. *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, 563–569. https://doi.org/10.1145/3328778.3366879

[17] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Long Beach, USA, 6000-6010.

[18] Min, T., & Lee, B. (2024). Comparison of Scoring Results for Physics Descriptive Items in GPT Language Models Based on Prompts. *New Physics: Sae Mulli, 74*(5), 504–514. http://doi.org/10.3938/NPSM.74.504

[19] Seong, J., & Shin, B. (2023). Exploring the Feasibility of Automatic Scoring of Written Test Using ChatGPT: Focusing on the World Geography Written Test. *Journal of the Association of Korean Geographers, 12(3)*, 415–432. https://doi.org/10.25202/JAKG.12.3.3

[20] Shin, B., Lee, J., & Yoo, Y. (2024). Exploring automatic scoring of mathematical descriptive assessment using prompt engineering with the GPT-4 model: Focused on permutations and combinations. *Journal of the Korean Society of Mathematical Education, 63(2)*, 187–207. https://doi.org/10.7468/mathedu.2024.63.2.187

[21] Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., & Zhou, D. (2022). Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*. https://doi.org/10.48550/arXiv.2203.11171

[22] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q., & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems 35*, New Orleans, USA, 24824–24837.

[23] Cohn, C., Hutchins, N., Le, T., & Biswas, G. (2024). A chain-of-thought prompting approach with llms for evaluating students' formative assessment responses in science. *Proceedings of the 38th AAAI conference on artificial intelligence, Vancouver, Canada, 38(21)*, 23182–23190. https://doi.org/10.1609/aaai.v38i21.30364

[24] Lee, G., Latif, E., Wu, X., Liu, N., & Zhai, X. (2024). Applying large language models and chain-of-thought for automatic scoring. *Computers and Education: Artificial Intelligence,* 6, 100213. https://doi.org/10.1016/j.caeai.2024.100213

[25] Weeda, R., Izu, C., Kallia, M., & Barendsen, E. (2020). Towards an assessment rubric for eipe tasks in secondary education: Identifying quality indicators and descriptors. *Proceedings of the 20th Koli Calling International Conference on Computing Education Research*, Koli, Finland, 1–10. https://doi.org/10.1145/3428029.3428031

[26] Hou, W., & Ji, Z. (2025). Comparing large language models and human programmers for generating programming code. *Advanced Science, 12(8)*, 2412279. https://doi.org/10.1002/advs.202412279

[27] Pandey, R., Singh, P., Wei, R., & Shankar, S. (2024). Transforming Software Development: Evaluating the Efficiency and Challenges of GitHub Copilot in Real-World Projects. *arXiv preprint arXiv:2406.17910*. https://doi.org/10.48550/arXiv.2406.17910

[28] Chang, L., & Ginter, F. (2024). Automatic short answer grading for Finnish with ChatGPT. *Proceedings of the 38th AAAI Conference on Artificial Intelligence*, Vancouver, Canada, 38(21), 23173–23181. https://doi.org/10.1609/aaai.v38i21.30363

[29] Hogg, R., McKean, J., & Craig, A. (2020). In*troduction to Mathematical Statistics* (8th ed., Global ed.). Pearson Education.

[30] Maier-Hein, L., Eisenmann, M., Reinke, A., Onogur, S., Stankovic, M., Scholz, P., Arbel, T., Bogunovic, H., Bradley, A., & Carass, A. (2018). Why rankings of biomedical image analysis competitions should be interpreted with care. *Nature communications, 9(1)*, 5217. https://doi.org/10.1038/s41467-018-07619-7

[31] Wikipedia contributors. (2025, September 4). *Kendall rank correlation coefficient—Wikipedia*, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Kendall_rank_correlation_coefficient&oldid=1309523623

[32] Seong, T. (2002). *Validity and Reliability* (2nd ed). Hakjisa.

[33] Park, M., & Park, Y. (2007). A New Measure of Agreement to Resolve the Two Paradoxes of Cohen's Kappa. *The Korean Journal of Applied Statistics, 20(1)*, 117–132. https://doi.org/10.5351/kjas.2007.20.1.117

[34] McHugh, M. (2012). Interrater reliability: The kappa statistic. *Biochemia Medica, 22(3)*, 276–282.

박민규

· 2021년 고려대학교 수학교육과(이학사)
· 2021년~현재 제주제일고등학교 교사
· 2024년~현재 한국교원대학교 대학원 컴퓨터교육과 석사과정

관심분야 : 프로그래밍 교육 및 평가, STEM, AI 교육, ML/DL 등
mingyu4796@naver.com

최현종

· 2005년 한국교원대학교 컴퓨터교육전공 (교육학 박사)
· 2006년~2021년 서원대학교 컴퓨터교육과 교수
· 2021년~현재 한국교원대학교 컴퓨터교육과 교수
· 2021년~현재 한국교원대학교 정보교육연구소 소장

관심분야 : 컴퓨터교육학, 정보교과교육, 인공지능 교육 등
chj@knue.ac.kr

부록

영문 초록, (영문)표 내용은 GPT-5을 이용하여 번역한 뒤, 저자가 최종 수정하여 완성하였습니다.

〈표 10〉 제안한 방식의 채점 프롬프트

구성 요소	내용
역할 (Role)	## Role and Objective You are a programming teacher in a Korean high school. Your mission is to grade students' explanations of highlighted code sections fairly and transparently, following the ##Instruction so the grading is clear and justifiable. Answer in Korean.
지시사항 (Instructions)	## Instructions Review the 〈Question〉 and the 〈Student's Answer〉 below. Based on the ##Detail rubric, assign a score and provide a justification. Think through the evaluation process step by step, but ensure the final output strictly follows ## Output Format. Step 1: You will receive a ##Question containing a code snippet with a //Highlighted Section//, along with a ##Student's Answer that explains what the student believes the purpose of the highlighted section. Step 2: Identify the key idea(s) that correspond to the ##Question. Step 3: Based on the key idea(s) in Step 2, Evaluate the student's answer using the criteria outlined in the ##Evaluation Dimensions section. Step 4: Based on your evaluation, assign a score according to the ##Detailed Rubric. Ensure strict adherence to the rubric guidelines. Step 5: Cross-reference your scoring decision with the provided ##Examples. If your assessment is inconsistent with the example patterns, revise your evaluation to ensure alignment with the established scoring standards. Step 6: Finalize the score and provide a clear justification for your decision. Return your evaluation using the exact JSON schema format specified in the requirements.
평가 차원 (Evaluation Dimensions)	## Evaluation Dimensions - Refer to the rubric to assign a score (0-6), considering: - **Correctness:** Is the explanation functionally accurate and aligned with the code? - **Unambiguity:** Is the description precise and is the terminology correct, avoiding multiple or contradictory interpretations and any incorrect wording? - **Abstraction:** Does the explanation give an appropriate high-level purpose, avoiding a line-by-line account? - Use the detailed rubric descriptions to determine the appropriate score and provide clear justification.
상세 루브릭 (Detailed Rubric)	## Detailed Rubric: { "Description": "Student responses varying across three continuous dimensions: abstraction, correctness, and Unambiguity. Initially, a multi-point three-tuple coding system was considered but deemed impractical due to grading complexity and the redundancy of distinguishing minor differences. Using this rubric, Construct the overall score by synthesizing how well or poorly the student's answer performs on each dimension. " "Rubric": [루브릭 내용]
출력 형식 (Output Format)	## Output Format Return this JSON structure: ```json { "Score": 〈integer 0-6〉, "Justification": 〈string explanation for the score, referring to the rubric and dimensions〉 }```
문항 (Question)	문항이 담고 있는 코드 조각이 제시됨.
채점 예시 (Scoring Examples)	총 6개의 실제 채점 사례(점수 및 근거 포함)를 제시함
학생 답안 (Student Answer)	평가 대상이 되는 학생의 서술형 답안

〈표 11〉 문항 1번의 교사 채점 예시

점수	학생 응답	채점 설명
6	가장 작은 값을 찾는 코드이다.	문제에 부합한 정답임.
5	이 코드는 arr이라는 집합에서 최솟값을 찾는 역할을 합니다.	핵심 아이디어는 정확하며, 다만 '집합'이라는 표현은 적절하지 않은 자료형임.
4	가장 작은 값을 정하는 코드이다.	코드의 목적을 잘 이해한 것으로 보이나, 무엇을 가장 작은 값으로 지정하느냐에 대한 여러 가지 해석이 있을 수 있으므로 다소 중의적임.
3	이 코드는 a값이 배열중에서 최소인 것이 되게 만드는 코드이다.	'최소인 것으로 되게 만든다'는 표현은 정답 여부를 헷갈리게 함. 핵심 아이디어인 최솟값을 "찾는" 과정을 파악했는지 그 해석이 애매모호하여 옳고 그름을 판별할 수 없음.
2	최소 값을 찾아서 출력하는 역할을 한다.	최솟값을 찾는다는 핵심 아이디어는 파악했으나, 한가지 이상의 필수적인 오류(주어진 영역에는 "출력"의 기능이 존재하지 않는다)가 있음.

점수	학생 응답	채점 설명
1	우선 a에 arr 리스트의 첫번째 값인 1을 저장하고, arr의 길이인 4번 반복하여 a의 값이 arr의 반복번째 값보다 클 때, a에다가 arr의 반복번째 값의 위치에 있는 값을 저장한다.	줄별 실행 과정을 나열하고 있어 매우 저수준의 추상화 단계를 나타냄.
0	이건 a를 출력하는거야	코드에 대한 아주 작은 설명이 있으나, 코드의 작동을 전혀 이해하지 못하고 아주 사소한 부분을 설명한 답안에 가까움.