JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 29 No. 4, pp. 31-38, April 2024
https://doi.org/10.9708/jksci.2024.29.04.031

Generative AI parameter tuning for online self-directed learning

Jin-Young Jun*, Youn-A Min**

*Student, Graduate School of Engineering, Hanyang Cyber University, Seoul, Korea
**Professor, Dept. of Applied Software Engineering, Hanyang Cyber University, Seoul, Korea

[Abstract]

This study proposes hyper-parameter settings for developing a generative AI-based learning support tool to facilitate programming education in online distance learning. We implemented an experimental tool that can set research hyper-parameters according to three different learning contexts, and evaluated the quality of responses from the generative AI using the tool. The experiment with the default hyper-parameter settings of the generative AI was used as the control group, and the experiment with the research hyper-parameters was used as the experimental group. The experiment results showed no significant difference between the two groups in the “Learning Support” context. However, in other two contexts (“Code Generation” and “Comment Generation”), it showed the average evaluation scores of the experimental group were found to be 11.6% points and 23% points higher than those of the control group respectively. Lastly, this study also observed that when the expected influence of response on learning motivation was presented in the ‘system content’, responses containing emotional support considering learning emotions were generated.

▸ **Key words**: e-Learning, Learning support tool, generative AI model, Hyper-parameters

[요 약]

본 연구는 온라인 원격교육에서 코딩 교육 활성화를 위해, 생성형 AI 기반의 학습 지원 도구 개발에 필요한 하이퍼 파라미터 설정을 제안한다. 연구를 위해 세 가지 다른 학습 맥락에 따라 하이퍼 파라미터를 설정할 수 있는 실험 도구를 구현하고, 실험 도구를 통해 생성형 AI의 응답 품질을 평가하였다. 생성형 AI 자체의 기본 하이퍼 파라미터 설정을 유지한 실험은 대조군으로, 연구에서 설정한 하이퍼 파라미터를 사용한 실험은 실험군으로 하였다. 실험 결과, 첫 번째 학습 맥락인 “학습 지원”에서는 실험군과 대조군 사이의 유의한 차이가 관찰되지 않았으나, 두 번째와 세 번째 학습 맥락인 “코드생성”과 “주석생성”에서는 실험군의 평가점수 평균이 대조군보다 각각 11.6% 포인트, 23% 포인트 높은 것으로 나타났다. 또한, system content에 응답이 학습 동기에 미칠 수 있는 영향을 제시하면 학습 정서를 고려한 응답이 생성되는 것이 관찰되었다.

▸ **주제어**: 원격교육, 학습 지원 도구, 생성형 AI, 하이퍼 파라미터

• First Author: Jin-Young Jun, Corresponding Author: Youn-A Min
*Jin-Young Jun (2022201681@hycu.ac.kr), Graduate School of Engineering, Hanyang Cyber University
**Youn-A Min (yah0612@hycu.ac.kr), Dept. of Applied Software Engineering, Hanyang Cyber University
• Received: 2024. 01. 22, Revised: 2024. 03. 15, Accepted: 2024. 03. 18.

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

I. Introduction

시·공간적 개방성, 경제성, 새로운 학습 경험 등 많은 이점으로 인해 이러닝 환경의 원격교육 과정에 참여하는 학습자가 늘어나고 있다. 그러나, 대면 학습환경보다 교수자 및 동료 학습자와의 상호작용이 자유롭지 못하고, 시스템의 낮은 편의성이나 개인 맞춤형으로 제공되지 않는 학습 컨텐츠 등은 학습자의 자기 주도 학습 활동을 어렵게 만들고 학습 만족도에도 부정적 영향을 주는 요인으로 지적되고 있다[1-3]. 이러한 연구를 볼 때, 학습자와 대화하듯 상호작용하며 대화의 맥락을 유지하고, 학습자에게 필요한 형식과 수준의 응답을 유도할 수 있는 생성형 AI는 원격교육 환경의 학습 지원 도구로 활용될 수 있는 큰 가능성이 있다.

온라인 코딩 교육 환경에서 생성형 AI를 학습 도구로 활용하여 학습한 학습자는 그와 같은 학습 도구를 활용하지 않은 학습자에 비해 논리적 사고력 및 자기효능감, 그리고 학습 동기 측면에서 더 긍정적인 영향을 받는다는 것을 확인한 기존 연구는 생성형 AI의 교육적 활용 가능성을 잘 보여주고 있다[4].

본 연구에서는 원격교육 환경에서 코딩 교육 활성화를 위한 학습 지원 도구로 생성형 AI를 활용하는 교수·학습 상황을 가정하고, 교수자를 보조하여 생성형 AI가 학습 맥락에 따른 적절한 응답을 생성할 수 있도록 선행연구의 고찰과 실험을 통해 교육적 활용에 적절한 하이퍼 파라미터 설정을 제안하고자 한다.

생성형 AI의 하이퍼 파라미터는 응답 문장의 생성에 영향을 주는 조건을 외부에서 조절하기 위한 변수인데, 따로 조절하지 않는 한 모델이 정한 기본값이 적용된다. 사용자가 생성형 AI를 사용할 때마다 여러 하이퍼 파라미터를 하나씩 조절하는 것은 상당히 불편한 작업이며, 하나 이상의 하이퍼 파라미터를 동시에 조절할 때 나타나는 복합 조절 효과도 예측하기 어려우므로 사용하기 편리한 학습 지원 도구를 개발하기 위해서는 사용 목적에 따른 조절 전략을 연구할 필요가 있다.

본 연구는 서로 다른 학습 맥락에 맞춤화된 생성형 AI의 하이퍼 파라미터 설정을 연구하는 것이 목적이며, 다음과 같은 방법으로 진행한다. 제2장에서는 하이퍼 파라미터의 동작 원리를 이해하기 위한 디코딩 전략과 대표적 생성형 AI 모델인 OpenAI의 ChatGPT API에서 지원하는 하이퍼 파라미터에 대해 알아본다. 또한, 하이퍼 파라미터 설정의 효과나 하이퍼 파라미터 최적화와 관련된 기존 연구를 고찰하여 시사점을 도출한다. 제3장에서는 기존 연구로부터 얻은 시사점을 기반으로 Python 코딩 학습 과정에서 학습 맥락에 따른 적절한 하이퍼 파라미터 설정을 제안하고, 학습 맥락별 실험 질문에 대한 응답을 평가한 후, 평가점수를 분석하여 제안한 설정의 효과성을 분석한다. 실험에 사용할 생성형 AI의 웹 인터페이스인 ChatGPT Playground의 기본 하이퍼 파라미터 설정(baseline)을 이용한 실험은 대조군으로, 제안한 설정에 따른 실험은 실험군으로 한다. 본 연구에서 말하는 학습 맥락은 “코드생성”, “주석생성”, 그리고 “학습 지원”으로 정의하며, 응답에 대한 평가 자료 수집을 위해 Python의 Web Framework인 Flask를 사용하여 웹 페이지 형식의 실험 도구도 구현한다. 마지막으로 제4장에서는 제3장에서 진행한 실험 결과를 토대로 원격교육 과정 중 코딩 교육 활성화를 위한 생성형 AI 기반 학습 지원 도구 개발 시 적용할 수 있는 하이퍼 파라미터 설정을 제안한다.

II. Preliminaries

1. Related works

1.1 Decoding strategy

생성형 AI의 디코딩 전략(Decoding strategy)이란 문장 생성에 필요한 토큰을 선택하는 방법을 의미하며, 가장 기본적인 Greedy search를 비롯하여 다양한 전략이 있다[5]. Greedy search는 문장 완성에 필요한 토큰을 선택할 때 무조건 확률이 가장 큰 토큰을 선택하는 방법이다. 이는 알고리즘이 단순하다는 장점이 있지만, 문장의 다양성이 낮고, 누적 확률이 최대가 아닌 문장도 최종 문장으로 생성되는 경우가 발생할 수 있다는 한계가 있다.

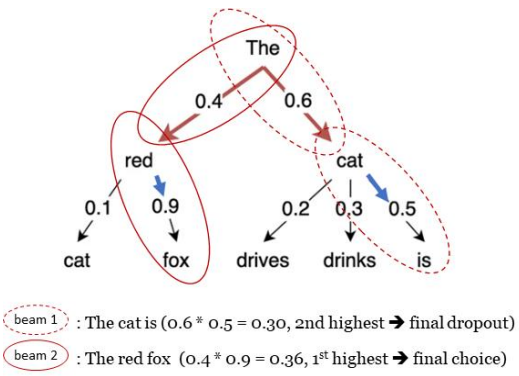


Fig. 1. Beam Search

Beam search는 확률 순서대로 지정된 수(beam width)만큼 문장의 후보(beam)를 확장해 나아간 후, 최종 단계에

서 가장 큰 확률을 가진 문장 하나를 출력하는 방법으로, 응답의 다양성을 높일 수는 있으나 반복되는 패턴이 나타날 수 있다. Fig. 1은 Beam width=2라고 가정했을 때 Beam search에 의해 생성되는 문장의 예시를 보여주는 그림이다. Beam width가 2라는 것은 후보 문장을 두 개 생성한다는 뜻이다. 처음 “The”라는 토큰으로 시작하여 다음에 이어질 수 있는 토큰 중 확률이 높은 순서대로 2개의 토큰을 선택하여 문장에 붙이면 “The red”와 “The cat”이라는 두 개의 후보 문장이 생성된다. 이후 문장이 완성될 때까지 동일한 과정을 반복하며 후보 문장에 토큰을 이어나가면 최종 단계에서는 “The red fox”와 “The cat is”라는 두 개의 후보 문장이 완성된다. 이들의 누적 확률을 비교하면 각각 0.36과 0.30이므로, 최종 문장으로는 누적 확률이 가장 큰 “The red fox”가 선택되는 것이다.

Temperature sampling은 Beam search의 문제인 패턴 반복성 해결을 위해 확률분포를 조절하는 방법이다. 즉, Temperature를 높이면 토큰의 확률분포가 균등해지므로 더 창의적이고 다양한 문장이 생성되며, 반대로 낮추면 확률분포의 모양이 날카로워져서 좀 더 흔하고 고정된 문장이 생성된다.

Top-k sampling은 확률이 높은 순서대로 고정된 k개의 후보 토큰 중 하나를 선택하는 전략이다. 이 방법은 맥락과 관계없이 k가 변하지 않으므로 일반적으로는 잘 사용하지 않는 부자연스러운 문장이 생성되기도 한다.

Top-p sampling(or nucleus sampling)은 고정된 k 대신 임계 확률 p를 넘는 누적 확률을 적용하므로 Top-k sampling의 문제를 어느 정도 완화하는 효과가 기대할 수 있다. Top-p를 0.1로 설정하면 상위 10%의 확률을 구성하는 토큰의 집합에서 하나의 토큰이 선택된다.

1.2 ChatGPT API Hyper-parameters

ChatGPT API는 응답 생성에 영향을 주는 하이퍼 파라미터로 ‘system content’, ‘temperature’, “max_tokens”, ‘n’, ‘frequency_penalty’, ‘presence_penalty’ 등을 제공한다[6]. ‘temperature’는 다양한 문장의 생성을 위해 토큰의 확률분포를 조절하고, ‘top_p’는 토큰의 확률밀도를 제어하여 샘플링되는 토큰의 결정성을 좌우한다. ‘n’은 응답의 개수이고, ‘max_tokens’는 단일 대화 세션에 포함될 수 있는 최대 토큰의 개수이며, ‘frequency_penalty’와 ‘presence_penalty’는 각각 등장 빈도와 이전에 등장한 토큰인지 아닌지에 따라 동일한 문장이 반복될 가능성과 다양성을 조절한다. ‘temperature’, ‘top_p’, ‘frequency_penalty’, ‘presence_penalty’는 모두 값이 커질수록 더 다양한 응답이 생성될 가능성을 높이며 값이 작아질수록 보수적이고 고정된 응답 생성을 유도한다. Table 1은 ChatGPT API의 하이퍼 파라미터를 간단히 표로 정리한 것이다.

Table 1. ChatGPT API Hyper parameters

Hyperparameter	Short Description
temperature	Control randomness for response diversity
top_p	Cumulative probability cutoff for token selection
max_tokens	Maximum number of tokens
frequency_penalty	Penalize new tokens based on their existing frequency in the text
presence_penalty	Penalize new tokens based on whether they appear in the text
n	Number of responses for each input message. Keep n as 1 to minimize costs.
stream	Reduce user waiting time when generating long response
stop	Number of sequence to stop generating further tokens
logit_bias	Control the likelihood of specified tokens in the response

1.3 Codex & EcoOptiGen

GPT(Generative Pre-trained Transformer) 기반의 코드생성 모델인 “Codex” 연구자들은 “HumanEval”이라는 데이터셋을 생성하여 “Codex”의 성능을 평가하고 그 결과를 데이터셋과 함께 공개하였다[7]. 공개된 “HumanEval”은 164개의 코드 문제(‘prompt’)와 정답(‘canonical-solution’) 등으로 구성된 dictionary 구조의 데이터셋이다. “Codex” 연구자들은 하나의 문제에 대해 여러 개의 응답을 생성하도록 하이퍼 파라미터를 설정했을 때 모델의 성능이 크게 높아진다는 것을 발견하고, 실험을 통해 관찰된 응답 개수(k)와 ‘temperature’ 간의 관계를 Fig. 2와 같이 시각화하여 보여주었다. Fig. 2에서 “Codex-S”는 “Codex”를 더 많은 코드로 학습시킨 모델을 말하고, ‘k’는 ChatGPT 하이퍼 파라미터 중 응답 개수를 뜻하는 ‘n’에 대응하는 하이퍼 파라미터로 볼 수 있다. Fig 2를 살펴보면, k와 best temperature 사이에는 양의 상관관계가 있음을 알 수 있다. 이는 ChatGPT 기반 학습 도구에서 n의 활용 가능성과 ‘temperature’를 설정할 때 ‘n’과의 관계를 고려할 수 있음을 시사하는 것으로 보인다.

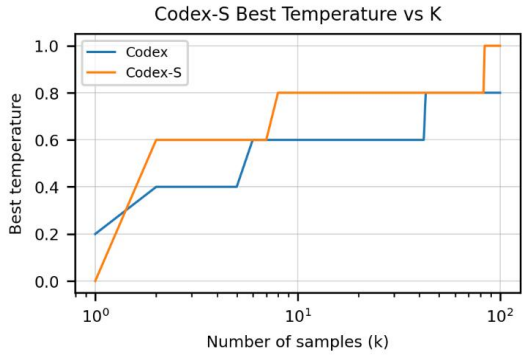


Fig. 2. No. of responses(k) vs temperature in Codex

한편, 생성형 AI의 하이퍼 파라미터 자동 최적화 프레임워크를 제안한 기존 연구에서는 Fig. 3과 같은 구조로 설계된 "EcoOptiGen"을 통해 미리 설정된 예산 범위 내에서 'model', 'max-tokens', 'temperature', 그리고 'top_p'의 최적값을 자동으로 탐색할 수 있다고 하였다[8].

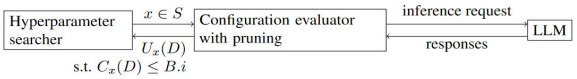


Fig. 3. EcoOptiGen Configuration

Fig. 3의 'Hyperparameter searcher' 모듈에서는 하이퍼 파라미터의 탐색공간(S)을 정의하고, 탐색공간의 균등분포로부터 임의의 설정($x \in S$)을 선택한다. 'Configuration evaluator with pruning' 모듈에서는 임의의 설정(x)을 적용한 질문(D)을 생성형 AI인 'LLM'에 보내고, 'LLM'으로부터 반환된 응답을 평가하여 평가점수(U_x)를 산출한다. "EcoOptiGen"은 이러한 일련의 과정에 드는 비용(C_x)이 추론예산($B.i$)을 넘지 않는 동안 같은 과정을 반복하며 평가점수(U_x)에 따라 업데이트해 가는 방법으로 최적의 하이퍼 파라미터 설정을 탐색하는 알고리즘이다. 연구자들은, 여러 개의 함수로 구성된 프로그램 단위와 함수 단위의 코드생성에 활용될 수 있는 데이터셋인 "APPS"와 "HumanEval", 수학 문제 풀이용 데이터셋인 "MATH", 그리고, 텍스트 요약을 위한 "XSum" 데이터셋을 기반으로 해당 프레임워크의 성능을 평가한 결과, 모든 데이터셋에 대하여 벤치마크 대비 우수한 성능을 보였다고 논문을 통해 밝히고 있으며, Python 코드생성과 수학 문제 풀이에 최적화된 하이퍼 파라미터의 설정을 Fig. 4와 같이 제시하였다. Fig. 4의 "HumanEval" Task 관련 실험 결과를 보면, 'n'=18, max_tokens=517, 그리고 'temperature' 대신 'top-p'를 사용하고, 그 값을 0.682로 설정했을 때 가장 우수한 성능이 관찰되었음을 알 수 있다.

Task	max_tokens	temperature_or_top_p	n
APPS	176	top_p: 0.982	15
HumanEval	517	top_p: 0.682	18
MATH	193	temperature: 1	26

Fig. 4. Optimized Hyper-parameters by EcoOptiGen

이상의 내용을 토대로, 본 연구에서는 다음 사항들을 고려하여 하이퍼 파라미터를 설정하고자 한다. 첫째 설정할 하이퍼 파라미터에는 'n'이 포함되어야 하고, 둘째, 'n'을 이용하여 'temperature' 값을 계산할 수 있는 함수를 추정한다. 셋째, 학습 맥락에 따라 "EcoOptiGen" 연구가 보여준 최적화된 설정을 참고한다.

다음 장에서는 학습 맥락에 따라 하이퍼 파라미터를 설정하고, 생성형 AI와 실험을 위한 대화를 주고받으며 생성형 AI의 응답을 평가하고 기록할 수 있는 실험 도구를 구현하며, 이를 이용하여 대조군과 실험군의 응답 평가점수(ScoreEval)를 실험 데이터로 수집한다. 그리고, 수집된 실험 데이터의 분석과 비교를 통해 두 집단 간의 평가점수에 유의한 차이가 있는지를 확인한다.

III. The Proposed Scheme

본 연구의 학습 맥락 중 "코드생성"과 "주석생성"에 대한 실험 질문은 "Codex" 관련 연구자들이 공개한 데이터셋인 "HumanEval"에서 "definition" 항목에 포함된 "docstring"과 "solution" 중 각각 무작위로 30개씩을 추출하여 사용하고, "학습 지원" 관련 실험 질문은 개발자 온라인 커뮤니티인 "Stackoverflow(S.Overflow)"에서 임의로 추출한 30개의 Python 3.x 관련 질문으로 한다[9]. 학습 맥락(CONTEXTS)별 실험 질문의 크기(SIZE)와 출처(SOURCE) 및 수집 항목(SAMPLE ITEM)은 Table 2와 같다. 본 논문에서 "Code. G"와 "Comm. G", "L. Support"는 각각 순서대로 "코드생성"과 "주석생성", 그리고 "학습 지원"을 말한다.

Table 2. Sampling of experiment questions

CONTEXTS	SIZE	SOURCE	SAMPLE ITEM
Code. G	30	HumanEval	Docstring in 'Definition'
Comm. G	30		Code in 'Solution'
L. Support	30	S.Overflow	Python 3.x Questions

실험 데이터에 기록되는 평가점수(ScoreEval)는 질문에서 요구하는 요구사항의 개수 대비 응답에서 충족한 요구사항의 비율에 최대 점수를 곱하고 이를 응답 개수로 나누어 산정한다. 평가점수는 식(2)에 의해서 계산한다.

$$ScoreEval = (\sum_{i=1}^{n} score_i) \times \frac{\max}{n}, \qquad 식(2)$$

$$score_i = \frac{cr}{qr}$$

식(2)에서 cr, qr은 각각 응답에서 충족한 요구사항의 개수와 질문에 포함된 요구사항 개수이고, $score_i$는 i번째 응답의 점수, n은 하이퍼 파라미터 'n'의 값을 말한다. 마지막으로 $\max$는 평가점수의 범위 조절 변수로, "코드생성"과 "주석생성"은 5, "학습 지원"은 1이다.

실험 도구는 'temperature', 'top-p', 'n', 'max-tokens'를 학습 맥락별로 설정할 수 있고, 질문 하나에 'n' 개의 응답을 출력하며, 응답에 대한 평가점수를 JSON 파일에 저장할 수 있다. 사용할 생성형 AI의 상세 모델은 대화식 상호작용에 최적화된 'gpt-3.5-turbo'로 한다.

다음은 학습 맥락에 따른 하이퍼 파라미터 설정 기준이다. (a) 영어보다 더 많은 토큰 수를 사용하는 한글을 고려하여 'max-tokens'는 모든 학습 맥락에서 기본 설정의 2배로 한다. (b) 'n'의 크기는 인지심리학에서 말하는 학습자의 작업기억 용량을 고려하여 학습 맥락에 따라 2 ~ 3의 범위에서 연구자가 실험을 통해 설정한다[10]. (c) "코드생성"과 "주석생성"에 대한 'temperature'는 제2장의 그래프(Fig. 2)에서 관찰된 데이터 포인트의 평균을 로그 함수의 기본형태에 적용하여 추정한 함수식에 의해 설정한다. 추정 로그 함수식은 식(1)과 같다.

$$0.125 \times \ln(1.123 \times n) + 0.303 \qquad 식(1)$$

Fig. 5는 식(1)의 함수로 추정한 temperature 값을 Fig. 2의 그래프와 비교할 수 있도록 중첩하여 시각화한 것이다. Fig. 5에서 "Codex"와 "Codex-S"의 그래프는 해당 선행연구에서 보여주었던 k와 best temperature 사이의 관계를 그대로 나타낸 것이고, "Assumption" 그래프는 본 연구에서 정의한 식(1)에 의해 추정된 n과 temperature 사이의 관계를 말한다.

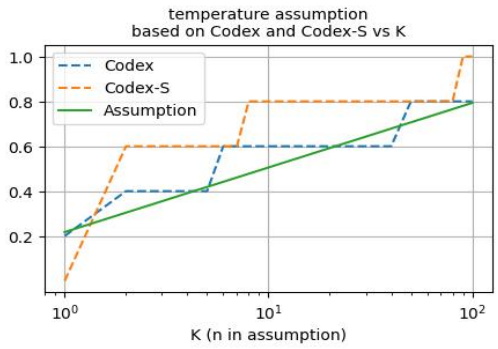


Fig. 5. 'temperature' assumption by n

(d) "학습 지원"은 비정형적인 다양한 질문을 위한 것이므로, 'temperature'는 식(1)의 함수를 통해 구해진 값과 "Codex"에서 샘플링에 사용한 값(0.8) 중 큰 값을 적용한다. (e) 'top-p'는 "EcoOptiGen" 이 제시한 최적화 결과를 참고하여 학습 맥락에 따라 달리 적용한다. 즉, "코드생성"의 'top-p'는 "HumanEval"에 대한 최적값인 0.682를 적용하고, "주석생성"의 'top-p'는 조금 더 비결정적인 문장이 생성되도록 "APPS"와 "HumanEval"에 대한 최적값의 평균을 적용한다. "학습 지원"의 'top-p'는 가장 다양한 응답이 생성될 수 있도록 'EcoOptiGen'의 최적값 범위에서 연구자가 실험을 통해 설정한다. (f) 프로그래밍 학습 과정에서는 토큰과 구문의 반복이 자주 발생하므로, frequency_penalty와 presence_penalty는 0으로 한다. (g) 코딩 학습이 재미있고 유쾌한 활동임을 시사하기 위해 "학습 지원"에 간단한 유머가 추가될 수 있도록 한다. 단, 조금 더 다양하고 창의적인 유머 생성을 위해 "학습 지원"에서 기대하는 응답보다 더 비결정적인 응답을 유도하는 하이퍼 파라미터를 설정한다. 마지막으로, (h) 'system content'는 비용과 대화 세션 유지에 영향을 주는 토큰 수를 고려하여 되도록 간단히 질문 상황과 기대하는 응답 형식을 제시하는 내용으로 한다. 학습 맥락별 하이퍼 파라미터 실험 설정을 요약하면 Table 3과 같으며, 학습 맥락별로 적용된 'system content'는 Table 4에 기술하였다.

Table 3. Experimental Hyper-parameter

Hyper-parameter	LEARNING CONTEXTS		
	Code. G	Comm. G	L. support
temperature	0.355	0.391	0.8
top_p	0.682	0.832	0.6
n	3	3	2
max_tokens	512	512	512

Table 4. System content per learning context

Learning contexts	system content
Code generation	You are a helpful Python 3 code generator. Give me Python code snippet for the following docstring and the code snippet is expected to run.
Comment generation	You are a helpful Python 3 comments generator. You will provide the user with line-by-line accurate comments for each line of the given python code. Your response should be formatted as following: """ # function name: addition, function parameters: a, b def addition(a,b): #return the value of a + b return a+b """
Learning support	You are a helpful Python 3 tutor having great sense of humor, experties and advanced teaching skills. Learners can be motivated by your humorous and sound advice based on teaching skills.

실험 도구로 실험을 시행하고, 수집된 실험 데이터를 기반으로 대조군과 실험군의 평가점수 평균이 유의한 차이가 있는지 확인하였다. 연구가설은 “실험군의 평가점수 평균은 대조군의 평가점수 평균보다 크다”이고, 귀무가설은 “실험군의 평가점수 평균은 대조군의 평가점수 평균과 같거나 작다”이다. 모수 검정인 대응표본 t 검정(단측)과 비모수 검정인 Wilcoxon test를 시행하여 두 검정의 결과를 모두 확인하는 방법으로 차이를 해석하였다. 학습 맥락별 분석 결과는 다음과 같다. “코드생성”에서는, 이상값으로 여겨진 1건의 표본을 제거한 후 총 29개 표본에 대해 검정한 결과, t 검정과 Wilcoxon 검정의 p-value가 각각 0.00015와 0.0008로 나왔다. 두 검정의 p-value가 모두 유의수준인 0.025보다 작았으므로, 실험군의 평가점수 평균이 대조군의 평균보다 유의하게 큰 것으로 해석하였다.

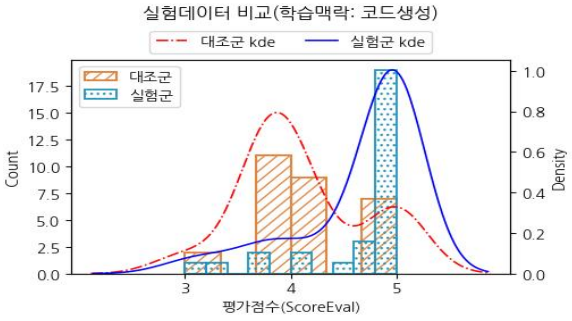


Fig. 7. Evaluation Scores(“Code generation”)

“코드생성”에 대한 평가점수를 히스토그램과 확률밀도 추정 그래프로 시각화한 Fig. 7을 보면, 대조군의 평가점수는 절반에 가까운 표본이 4점 미만이고, 나머지 절반 정도가 4~5점 사이 반면, 실험군은 표본 대부분이 4~5점 사이이며, 그중에서도 모든 요구조건을 충족하여 5점으로 평가된 표본이 가장 많았다.

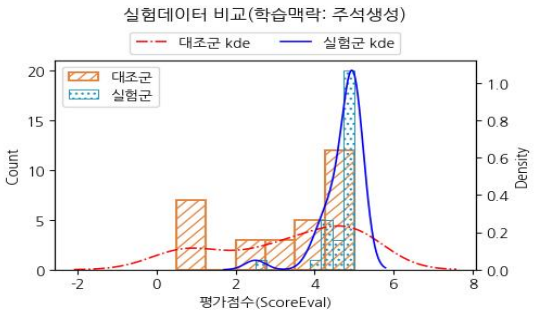


Fig. 8. Evaluation Scores(“Comment generation”)

“주석생성” 에서는 이상값이 발견되지 않았고, t 검정과 Wilcoxon 검정의 p-value는 각각 0.00017과 0.00023으로, 유의수준 0.025보다 작았다. 따라서 “코드생성”의 결과와 마찬가지로 실험군의 평가점수 평균이 대조군의 평균보다 유의하게 큰 것으로 나타났다. “주석생성” 실험의 평가점수 도수분포와 확률밀도 추정(kde) 그래프인 Fig. 8을 보면, 대조군에서는 약 절반 정도의 표본이 4점 미만이고, 나머지 절반만 5점으로 평가되었지만, 실험군에서는 표본 대부분이 5점으로 평가되었다.

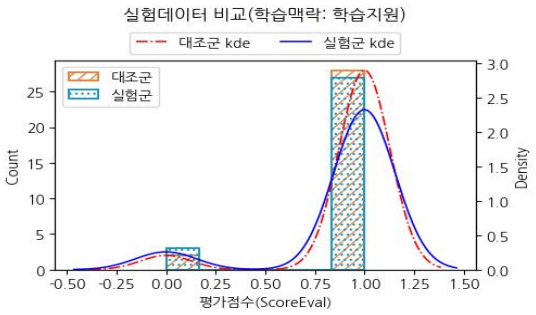


Fig. 9. Evaluation Scores(“Learning support”)

“학습 지원”에 대한 실험 데이터 검정 결과, t 검정과 Wilcoxon 검정의 p-value는 각각 0.668과 0.672로 모두 유의수준(0.025)보다 컸으므로, 귀무가설을 채택하여 실험

군의 평가점수 평균은 대조군의 평가점수 평균과 같거나 작은 것으로 해석하였다. “학습 지원”에 대한 응답은 내용 중심으로 평가되었으므로, 문장의 미완성 정도는 평가점수에 반영되지 않았다. 그러나, 미완성 응답이 한 건도 발생하지 않았던 실험군과는 달리 대조군에서는 총 질문 30개 중 10개에 대한 응답에서 미완성 문장이 반환되었다. 응답 내용 중 한 번이라도 질문에 대한 올바른 답변이 포함된 경우 요구사항을 충족한 것으로 평가하였기 때문에 문장의 미완성 정도가 컸음에도 불구하고 대조군과 실험군 간의 평가점수 평균에 유의한 차이가 없다고 나타난 것으로 보인다. Fig. 9는 평가점수의 범위가 0 ~ 1이었던 “학습 지원”에 대한 대조군과 실험군의 평가점수 도수분포와 확률밀도 추정을 나타낸 그래프이다. 문장의 미완성 정도를 고려하지 않고, 응답 내용 중 한 번이라도 올바른 답변이 포함되면 요구사항을 충족한 것으로 본 평가 기준에 따라 평가된 “학습 지원” 실험 데이터에서는 대조군과 실험군 모두에서 대부분의 응답이 요구조건을 충족한 것으로 평가되었다.

한편, 유머 생성을 위해서는 질문에 대한 응답보다 더 다양한 문장이 생성될 수 있도록 기본 설정을 적용하되, 짧은 유머 생성을 위해 ‘max_tokens’만 기본 설정의 0.5배로 하였다. 실험에서 생성된 유머들은 모두 Python 관련 범위를 벗어나지 않았고 유머임을 알 수 있는 문체의 완성된 문장이 형성되었으며, gpt-3.5-turbo 기준으로 폭력성, 인권 차별 등과 같은 부정적 표현은 관찰되지 않았다.

학습 맥락별 대조군과 실험군의 평가점수 평균을 백분율로 비교하면 Table 5와 같다. “코드생성”과 “주석생성”에서는 실험군의 평가점수 평균이 대조군보다 각각 11.6% 포인트, 23% 포인트 높았고, “학습 지원”에서는 실험군과 대조군 사이의 유의한 차이가 관찰되지 않았다.

Table 5. Results of Experiment

Learning contexts	Mean ScoreEvals		Effects
	baseline	experiment	
Code. G	81.8%	93.4%	11.6% point ↑
Comm. G	71.2%	94.2%	23% point ↑
L. Support	93.33%	90%	No significant difference

IV. Conclusions

복잡한 학습 영역인 프로그래밍 언어를 처음 학습하는 학습자의 질문은 비정형적이고 추상적일 수 있다는 것과 탐구할 수 있는 풍부한 인지적 학습환경의 제공을 중요하게 여기는 구성주의 학습이론에 비추어 볼 때, 온라인 원격교육 환경의 학습자에게는 질문에 대한 응답 내용을 충분히 검토하고 보다 구체적인 질문으로 발전시켜가는 과정 자체가 지식 구성의 중요 요인이 될 수 있다[11]. 따라서, ‘max_tokens’는 본 연구에서 제시한 설정값을 적용하여 응답이 중간에 끊기지 않도록 유도하는 것이 더 바람직할 것으로 생각된다.

마지막으로, 실험 중 관찰된 특이한 사항으로는 ‘system content’ 설정이 응답의 적절성뿐만 아니라 정서적 지지 표현에도 영향을 준다는 것이었다. 예를 들면, “You are a helpful Python 3 tutor”와 같이 system의 역할을 간단한 명사구로 제시한 경우와는 다르게, Table 4의 “학습 지원(Learning support)”에서와 같이 “system content”에 학습 동기에 미칠 수 있는 영향을 함께 제시했을 때, 마치 실세계의 교수자가 학습에 대한 학습자의 어려움을 공감하고 배려하는 듯한 문체로 응답이 형성되는 것이 관찰되었다. 그러므로 ‘system content’는 학습자의 학습 수준에 대한 배려, 격려 등과 같이 ‘정확성’ 기준으로는 측정하기 어려운 정서적 지지를 표현하는 응답 형성을 유도하기 위한 목적으로도 활용될 수 있을 것으로 보인다.

본 연구를 통해 제안한 하이퍼 파라미터 설정으로 “코드생성”과 “주석생성” 맥락에서 기본 설정보다 더 높은 평가점수를 받을 수 있고, 학습에 재미를 주는 유머와 정서적 지지를 담은 응답도 유도할 수 있음을 확인하였다. 온라인 코딩 교육 환경의 학습 지원 도구 개발 시 본 연구의 하이퍼 파라미터 설정을 적용한다면, 학습자 맞춤형 응답을 유도하여 학습자가 능동적이고 주도적인 자세로 문제를 해결해 나가는 과정을 경험하게 할 수 있을 것이다. 이는 교수·학습적 측면에서 볼 때, 학습자 주도의 학습 과정 전개를 가능하게 하며 학습 동기를 유발하고 단계적 학습에 의한 학습 연계를 유연하게 할 수 있으며, 중도 탈락의 위험도 감소시킬 수 있을 것이다.

REFERENCES

[1] Gyeng Suk Jeong,“College Students’ Recognitions toward Online Tutoring and Factor Analysis on Class Satisfaction,” Journal of Extra-curricular Research, Vol. 1, No. 3, pp.25-41, 2020

[2] Seo Il Bo, “Differences in Perceptions of Non-face-to-face Liberal Arts Classes at Universities and their Effect on Satisfaction,” Korean Journal of General Education, Vol. 15, No. 6, pp.288-299, 2021

[3] Lee Ssang-cheol, Kim Jeong-a, “Factors that affect student satisfaction with online courses,” The Journal of Educational Administration, Vol. 36, No. 2, pp.115-138, 2018

[4] Ramazan Yilmaz, Fatma Gizem Karaoglan Yilmaz, "The effect of generative artificial intelligence (AI)-based tool use on student’s computational thinking skills, programming self-efficacy and motivation," Computers and Education: Artificial Intelligence, Vol. 4, 2023, 100147, DOI:10.1016/j.caeai.2023.100147

[5] Fabio Chiusano, Two minutes NLP – Most used Decoding Methods for Language Models, *https://medium.com/nlplanet/two-minutes-nlp-most-used-decoding-methods-for-language-models-9d44b2375612*

[6] ChatGPT API Reference, *https://platform.openai.com/docs/api-reference/chat*

[7] Mark Chen et al, “Evaluating Large Language Model Trained on Codex,” arXive preprint, arXiv:2107.03374v2 [cs.LG], 2021

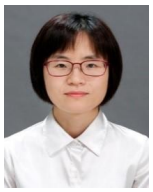
[8] Chi Wang, Susan Xueqing Liu, Ahmed H. Awadallah, “Cost-Effective Hyperparameter Optimization for Large Language Model Generation Inference,” arXive preprint, arXiv:2303.04673v2 [cs.CL], 2023

[9] Stackoverflow, *https://stackoverflow.com/questions*

[10] Randal W. Engle, “Working Memory Capacity as Executive Attention,” Current Directions in Psychological Science, Vol. 11, No. 1, pp. 19-23, 2002, DOI:10.1111/1467-8721.00160

[11] Wulf. Tom, Constructivist approaches for teaching computer programming, Proceedings of the 6th conference on Information technology education, pp.245-248, SIGITE ’05 Newark NJ, USA, Oct. 2005, DOI: 10.1145/1095714.1095771

Authors



Jin-Young Jun received the B.S. degree in Computer Science from Sungshin Women’s University, Korea, in 2003, and M.S. degree in e-Learning from Korea National Open University in 2020.

Jun is currently pursuing second M.S. degree in Mechanical & IT Convergence Engineering from Graduate School of Engineering, Hanyang Cyber University, Seoul, Korea. She is interested in IoT, artificial intelligence and machine learning.

Youn-A Min Received a doctorate in computer engineering from Dongguk University. She served as a professor at Gachon University from 2016 to 2019, and has been working as a professor in the Department of Applied Software Engineering at Hanyang Cyber University since 2020. Dr. Min has been working as a professor in the Department of Applied Software Engineering at Hanyang Cyber University in Seoul since 2020, and is also a professor at the Graduate School of Mechanical and IT Convergence. She is interested in blockchain and blockchain-based artificial intelligence security.