JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 31 No. 10, pp. 159-174, October 2026
https://doi.org/10.9708/jksci.2026.31.10.159

Research Trends in Korean and International Studies on Vibe Coding: A Text Mining Approach

Tae-Ho Min*

*Teacher, Bora High School, Yongin, Korea

[Abstract]

This study comparatively analyzed research trends and topics in 83 Korean and international academic articles related to vibe coding. To this end, TF-IDF-based keyword time-series analysis and LDA-based topic modeling were conducted using the English abstracts of the articles collected from the Web of Science and the Korea Citation Index. The results showed that international research on vibe coding expanded from an initial focus on applications in specific domains and the feasibility of vibe coding to a broader focus on natural language-based software development approaches, code generation and development automation, and issues of responsibility and governance. International research topics included software development in medical and clinical contexts, AI-assisted educational tool development, and responsible software development. In contrast, Korean research focused on practical applications in educational and learning contexts, including digital tool development, learner experiences, and teachers' design and development activities. Korean research topics included the development of learning tools for specific educational contexts, such as music, career, language, and elementary education, as well as digital and data-driven research and AI-scaffolded learning. Based on these results, vibe coding research was discussed as an extension of programming abstraction.

▸ **Key words**: Vibe coding, Programming abstraction, Large language models, TF-IDF, Topic modeling

[요 약]

본 연구는 국내외 바이브 코딩 관련 학술 논문 83편을 대상으로 연구 동향 및 주제를 비교·분석했다. 이를 위해 Web of Science와 KCI에서 수집한 문헌의 영문 초록을 대상으로 TF-IDF 기반 키워드 시계열 분석과 LDA 기반 토픽 모델링을 수행했다. 연구 결과, 국제 연구는 특정 분야 적용 사례와 개발 가능성 탐색에서 출발해 자연어 기반 소프트웨어 개발 접근법, 코드 생성 및 개발 자동화, 책임성 및 거버넌스 등으로 연구 관심이 확장되었으며, 의학·임상 분야 소프트웨어 개발, AI 기반 교육 도구 개발, 책임 있는 소프트웨어 등의 주제가 도출되었다. 국내 연구는 교육 및 학습 맥락을 중심으로 디지털 도구 개발, 학습자 경험, 교사의 설계 및 개발 활동 등 현장 적용에 초점을 두었고, 음악·진로·언어·초등교육 등 구체적 교육 맥락의 학습 도구 개발과 디지털·데이터 기반 연구, AI 기반 학습 스캐폴딩 등의 주제가 나타났다. 이상의 결과를 바탕으로 프로그래밍 추상화의 연장선으로서 바이브 코딩 연구를 논의했다.

▸ **주제어**: 바이브 코딩, 프로그래밍 추상화, 대형 언어 모델, TF-IDF, 토픽 모델링

• First Author: Tae-Ho Min, Corresponding Author: Tae-Ho Min
*Tae-Ho Min (ty3370@dankook.ac.kr), Bora High School
• Received: 2026. 08. 18, Revised: 2026. 09. 18, Accepted: 2026. 09. 21.

Copyright © 2026 The Korea Society of Computer and Information
http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945
www.kci.go.kr

I. Introduction

생성형 인공지능(Generative artificial intelligence; 이하 생성형 AI)과 대형 언어 모델(Large language model; 이하 LLM)의 발전은 소프트웨어 개발 방식에도 새로운 변화를 불러왔다. LLM을 활용한 코드 생성과 소프트웨어 공학 적용에 관한 연구는 코드 생성, 테스트, 디버깅, 문서화 등 다양한 개발 활동으로 확장되었으며, 관련 기술의 가능성과 한계에 관한 논의도 이루어졌다[1]. 이러한 변화 속에서 사용자가 프로그래밍 언어를 직접 작성하는 대신 자연어로 자신의 요구와 의도를 표현하고, LLM이 이를 바탕으로 코드와 소프트웨어를 생성하는 방식인 바이브 코딩(Vibe coding)이 새로운 개발 방식으로 부상했다[2]. 바이브 코딩은 전문적인 프로그래밍 지식이 부족한 사용자의 소프트웨어 개발 참여를 확대하는 한편, 전문 개발자에게도 코드 작성과 구현 과정의 일부를 시스템에 위임하는 방식을 제시했다[3].

바이브 코딩에 관한 연구는 단일 영역이나 관점에 한정되지 않고 다양한 맥락에서 이루어지고 있다. 예컨대 의학 분야에서의 소프트웨어 및 도구 개발[4, 5], 교육용 디지털 도구의 개발과 활용[6-8], 데이터 분석을 위한 소프트웨어 개발[9] 등 여러 분야에 걸쳐 연구가 이루어졌다. 바이브 코딩의 활용 주체 역시 학습자, 교육자, 도메인 전문가 및 소프트웨어 개발과 관련된 다양한 사용자 맥락에서 논의되었다[10, 11]. 이는 바이브 코딩이 특정 기술이나 개별적인 활용 사례만으로 포괄하기 어려운 다양한 연구 관심과 적용 영역을 형성하고 있음을 보여준다.

이러한 맥락에서 바이브 코딩 관련 연구를 종합적으로 검토하려는 시도가 수행되었다[3, 12]. 그러나 기존의 문헌 분석은 학술 논문과 실무 자료를 모두 고려하는 방식으로 이루어졌으며, 새롭게 형성되고 있는 바이브 코딩 관련 학술 연구 자체가 시간의 흐름에 따라 어떻게 변화하고 있는지, 그리고 국내외 연구에서 어떠한 주제들이 주요하게 형성되고 있는지를 종합적으로 분석한 연구는 찾아보기 어렵다. 특히 바이브 코딩 연구는 다양한 도메인과 적용 맥락에서 전개되고 있으므로, 개별 연구의 내용이나 특정 적용 사례를 중심으로 이해하기보다 관련 학술 문헌 전반을 대상으로 연구의 시간적 변화와 주제 구조를 파악할 필요가 있다.

이에 본 연구는 국내외 바이브 코딩 관련 연구 동향과 주제를 분석해 바이브 코딩 연구의 전반적인 지형을 탐색적으로 파악하고자 했다. 이를 위해 국내외 학술 데이터베이스에서 수집한 바이브 코딩 관련 문헌을 TF-IDF 키워드 추출 및 LDA 토픽 모델링으로 분석하고 시기별 연구 동향과 주요 연구 주제를 도출했다. 본 연구의 연구 문제는 다음과 같다.

1. 국내외 바이브 코딩 관련 연구 동향은 시간의 흐름에 따라 어떻게 나타나는가?
2. 국내외 바이브 코딩 관련 연구에서 나타나는 주요 연구 주제는 무엇인가?

II. Preliminaries

바이브 코딩은 표면적으로 최근 등장한 새로운 현상처럼 보이지만, 그 핵심적인 문제의식은 소프트웨어 공학 및 인간-컴퓨터 상호작용 연구에서도 다루어졌다. 즉 프로그래밍의 구체적인 구현 절차를 인간이 직접 작성해야 하는가, 또는 사용자의 의도나 원하는 결과를 특정 시스템이 프로그램으로 변환할 수 있는가의 문제이다. 이러한 질문은 비프로그래머 프로그래밍(Non-programmer programming)을 비롯해 최종 사용자 컴퓨팅(End-user computing), 시연에 의한 프로그래밍(Programming by demonstration), 로우코드/노코드(Low-code/no-code) 개발에 이르기까지 다양한 접근 방식으로 연구되었다. 따라서 바이브 코딩은 기존의 프로그래밍 패러다임과 단절된 새로운 현상이라기보다, 구현 세부 사항을 점차 시스템에 위임하고 사용자가 높은 수준에서 의도를 표현할 수 있도록 발전해 온 장기적인 기술적 변화의 최근 단계로도 볼 수 있다. 본 연구의 이론적 배경에서 이러한 변화의 전체 과정을 포괄적으로 검토하는 것은 아니다. 바이브 코딩을 이해하는 데 직접적으로 관련되는 관점, 즉 사용자가 프로그램의 구현 세부 사항을 얼마나 직접 다루어야 하는가와 사용자의 의도를 시스템이 어떻게 프로그램으로 변환하는가에 초점을 맞추어 관련 선행 연구의 흐름을 살펴본다. 또한 바이브 코딩은 비프로그래머뿐만 아니라 전문 개발자도 활용하는 방식이라는 점을 고려해, 본 장에서는 특정 사용자 집단의 변화보다는 프로그래밍 추상화(Abstraction)와 구현 세부 사항의 위임이라는 관점에서 몇 가지 선행 연구를 검토한다. 여기서 프로그래밍 추상화는 프로그래밍 언어 자체의 추상화 계층보다는, 사용자가 프로그램의 구현 세부 사항을 직접 다루는 부담을 줄이고 보다 높은 수준에서 자신의 의도를 표현하는 사용자 수준(User-level) 추상화에 초점을 둔다.

www.kci.go.kr

1. A brief history of user-level programming abstraction

컴퓨터 프로그래밍의 초기 발전 과정에서 코딩은 주로 전문적인 프로그래머의 영역으로 인식되었다. 프로그래머는 특정 프로그래밍 언어와 컴퓨터의 작동 원리를 이해하고 이를 바탕으로 명시적인 명령어를 작성했다. 그러나 컴퓨터의 활용 범위가 확대되면서 전문적인 프로그래밍 지식을 갖추지 않은 일반 사용자도 컴퓨터를 이용하여 자신의 문제를 해결할 필요성이 증가했다. 이러한 문제의식은 1970년대의 연구에서도 발견된다. 예컨대 Miller[13]는 프로그래밍 언어의 표현 방식과 인간의 문제 해결 방식 사이의 관계를 연구하면서, 인간이 컴퓨터에 지시하기 위해 반드시 특정 프로그래밍 언어의 형식적 문법을 학습해야 하는지에 대한 질문을 던졌다. 이는 프로그래밍 언어의 표현 방식과 인간의 문제 해결 방식 사이의 관계를 탐색한 초기 연구의 한 사례로 볼 수 있다[14].

이와 유사하게 Onorato & Schvaneveldt[15]는 프로그래머와 비프로그래머가 사람과 컴퓨터에 절차를 설명하는 방식을 비교함으로써 일반 사용자들이 가지고 있는 자연어 기반 시스템에 대한 기대를 확인했으며, 초보 프로그래머가 경험하는 어려움을 분석한 Ebrahimi[16]의 연구 역시 프로그래밍 언어의 문법적·구조적 요구가 사용자의 자연스러운 문제 해결 방식과 반드시 일치하지 않는다는 점을 보여주었다. 이러한 초기 연구들은 적어도 부분적으로, 프로그래밍의 어려움이 단순히 컴퓨터에 원하는 일을 설명하는 것 자체에서 발생하는 것이 아니라, 인간의 의도를 컴퓨터가 요구하는 형식적 표현으로 변환하는 과정에서도 발생할 수 있음을 시사한다. 이는 사용자가 컴퓨터의 요구에 맞추어 프로그래밍 언어를 학습해야 하는 전통적 관점에서 벗어나, 컴퓨터가 사용자의 자연스러운 문제 표현 방식에 가까워져야 한다는 내추럴 프로그래밍(Natural programming) 관점과도 일맥상통한다[17].

1980년대 이후 개인용 컴퓨터와 정보시스템이 사회 조직 전반으로 확산되면서, 일반 사용자가 정보시스템을 직접 활용하고 필요에 따라 프로그램을 개발 또는 수정하는 최종 사용자 컴퓨팅(End-user computing; 이하 EUC) 연구가 발전했다. Rivard & Huff[18]는 EUC의 성공 요인으로 일반 사용자에 의한 직접적인 응용 프로그램 개발을 제시했으며, 그 연장선에서 Brancheau & Brown[19]은 특정 도메인의 업무 지식과 경험을 가진 비프로그래머가 소프트웨어 개발의 주체가 될 수 있다는 점을 EUC의 강점으로 보았다. Fischer et al.[20]은 EUC의 관점을 메타 설계(Meta-design)로 발전시켜, 시스템을 처음부터 완성된 상태로 제공하는 대신 사용자가 자신의 필요에 따라 시스템을 변화시키고 확장할 수 있도록 설계하는 접근을 제안했다. Mørch et al.[21]도 이와 유사하게 일반 사용자가 저수준 코드를 직접 작성하지 않고도 높은 수준의 구성 요소를 조합하여 소프트웨어를 개발하는 방법을 연구했다.

한편, 일반 사용자가 프로그램의 구현 세부 사항을 직접 기술하지 않고도 자신의 의도를 시스템에 전달할 수 있도록 하는 또 다른 중요한 연구 흐름은 시연에 의한 프로그래밍(Programming by demonstration; 이하 PBD)이었다. 이 접근법에서는 사용자가 전통적인 프로그래밍 언어로 명령어를 직접 작성하는 대신, 원하는 작업을 수행하는 예시나 행동을 시스템에 보여주고 시스템이 이를 일반화하여 프로그램을 생성하도록 했다. PBD 관련 연구는 1980년대 이후 다양한 형태로 발전했는데, 대표적으로 PBD를 지원하는 시각적 환경에 관한 연구[22], PBD를 적용한 로봇 프로그래밍에 관한 연구[23], 비프로그래머의 PBD 활용에 관한 연구[24] 등이 이루어졌다. 나아가 Chen[25]은 PBD가 사용자의 행동을 단순히 모방하는 것을 넘어 그 행동 뒤에 존재하는 상위 수준의 목적과 의도를 추론하는 방향으로 발전할 가능성을 보여주었으며, Castelli et al.[26]은 이를 구체화하여 PBD에서 시스템의 결과를 사용자가 다시 수정하고, 이러한 수정 정보를 다시 프로그램 생성 과정에 반영하는 반복적 피드백 과정을 연구했다.

프로그래밍에 요구되는 구현 세부 사항을 줄이고 상위 수준에서 소프트웨어를 구성하려는 문제의식은 로우코드/노코드(Low-code/No-code; 이하 LCNC) 개발에서도 나타났다. LCNC는 전문 개발자가 아닌 사용자도 시각적 구성 요소, 작업 흐름, 모델 등을 이용해 소프트웨어를 개발할 수 있도록 하는 접근이다[27]. LCNC 플랫폼의 등장으로 일반 사용자가 실제 개발과 배포에 참여할 수 있게 되면서, EUC에서 나타난 사용자 주도 접근은 소프트웨어의 구성과 개발, 배포까지 그 범위를 확장하게 되었다[28]. Viljoen et al.[29]은 LCNC가 단순히 개발 편의성을 높이는 기술이 아니라 누가 소프트웨어를 개발할 수 있는가와 소프트웨어 개발을 조직 내에서 어떻게 분담할 것인가를 변화시키는 기술적·조직적 접근으로 보았으며, Rafi et al.[30] 또한 LCNC의 발전을 전문 프로그래머와 비프로그래머의 개발 참여 방식 자체를 변화시키는 과정으로 해석했다.

이상의 선행 연구는 사용자의 의도를 구현 가능한 프로그램으로 변환하는 과정에서 인지적·기술적 장벽을 낮추기 위한 다양한 접근법이 시도되었음을 보여준다. 각 접근법은 프로그래밍에서 사용자가 직접 다루어야 하는 구현의 세부 사항을 줄이고, 더 높은 수준에서 자신의 의도를 표

www.kci.go.kr

현할 수 있는 방식들을 제시했다. 그러나 이러한 접근법은 대체로 사전에 정의된 구성 요소, 제한된 조작 환경, 예시 또는 특정한 중간 표현 등을 요구했으며, 사용자의 개방형 자연어 의도를 별도의 구조화 과정 없이 프로그램으로 변환하는 데에는 한계가 있었다. 이러한 한계는 자연어를 이해하고 코드로 변환할 수 있는 생성형 AI의 등장과 함께 새로운 방식으로 해소될 가능성을 갖게 되었으며, 이는 바이브 코딩이 기존의 프로그래밍 추상화와 구별되는 중요한 지점이다. 다만 이러한 구별은 바이브 코딩이 비프로그래머만을 위한 기술이라는 의미는 아니다. 바이브 코딩은 전문 개발자에게도 구현 세부 사항의 일부를 시스템에 위임할 수 있는 새로운 방식으로 활용될 수 있으며, 이러한 점에서 기존의 프로그래밍 추상화 연구보다 넓은 사용자 범위를 갖는다.

2. The emergence of LLMs and vibe coding

ChatGPT, Gemini, Claude 등 LLM의 등장은 프로그래밍에서 추상화가 이루어지는 방식, 특히 사용자의 의도를 실행 가능한 프로그램으로 변환하는 과정에서 인간이 직접 부담해야 하는 형식화의 정도를 변화시켰다. 이전의 프로그래밍 추상화 접근법에서 사용자는 자연어로 생각하더라도 시각적 구성 요소, 작업 흐름, 예시 또는 사전에 정의된 기능 단위 등을 통해 자신의 의도를 시스템이 이해할 수 있는 형태로 구체화해야 했다. 이와 달리 LLM을 활용한 프로그래밍은 사용자의 자연어 표현 자체를 프로그램 생성의 입력으로 활용할 수 있게 함으로써 형식 변환 과정의 상당 부분을 시스템 내부로 이동시켰다[31].

이러한 맥락에서 2025년 바이브 코딩이라는 용어가 등장했으며, LLM을 활용한 새로운 형태의 개발 방식이 명시적으로 논의되기 시작했다. 바이브 코딩은 사용자가 자연어를 통해 원하는 기능과 결과를 설명하고 LLM이 생성한 프로그램을 실행한 후 그 결과에 대한 추가적인 요구사항을 전달하는 방식으로 소프트웨어를 개발하는 접근 방법이다[2, 32]. Meske et al.[2]은 바이브 코딩을 단순한 코드 생성 기술이 아닌 소프트웨어 개발에서 의도 매개(Intent mediation) 방식의 재구성으로 개념화했다. 전통적인 프로그래밍에서는 개발자가 사용자의 요구를 코드로 직접 변환하는 과정 자체를 담당하지만, 바이브 코딩에서는 LLM이 자연어로 표현된 의도와 실행 가능한 프로그램 사이를 매개한다는 것이다. 따라서 바이브 코딩의 등장으로 나타난 중요한 변화는 단순히 코드 작성의 일부가 자동화되었다는 데 있는 것이 아니라, 프로그램을 구성하는 과정에서 사용자의 의도가 코드로 변환되는 방식 자체가 변화했다는 점에 있다.

다만 이러한 변화가 프로그래밍 과정에서 인간의 역할을 제거하는 것은 아니다. Oertel et al.[33]은 GitHub Copilot이 생성한 여러 해결책을 분석하면서, LLM이 하나의 프로그램을 생성하는 것보다 여러 생성 결과를 인간이 검토하고 선택하는 과정이 중요하다는 점을 보여주었다. Tambon et al.[34]은 LLM이 생성한 코드에서 오류가 발생할 수 있음을 실증함으로써 프로그래밍 과정에서 인간의 검증이 불필요해지는 것은 아니며, 생성된 프로그램의 정확성을 판단하고 오류를 발견하는 과정이 중요함을 주장했다. 이상의 연구들은 인간의 역할이 프로그램을 처음부터 직접 작성하는 것에서 생성된 결과를 검토·선택·수정하는 활동으로 일부 이동할 수 있음을 보여준다. 이는 과거 PBD 연구에서 제안된 사용자의 수정과 피드백을 프로그래밍에 다시 반영하는 반복적 상호작용 구조가[26] 고차원적 의도 표현과 프로그램 생성 과정에 적용된 것으로도 볼 수 있다.

최근에는 바이브 코딩을 실제 개발에 사용하는 경험적 연구가 수행되었다. Pesce & Cheungpasitporn[4]은 신장학 교육을 위한 대화형 학습 도구를 바이브 코딩을 통해 개발하는 사례를 제시했으며, Meyer[9]는 Omics 데이터 분석을 위한 소프트웨어를 바이브 코딩으로 개발하는 과정을 보고했다. Chow & Ng[35]은 교육 분야에서 바이브 코딩을 활용해 비프로그래머가 교육용 기술을 개발하는 가능성을 논의했고, Ng & Chow[36]은 실제 교육용 소프트웨어 개발 사례를 통해 이를 구체화했다. 이처럼 현재의 바이브 코딩 연구는 다양한 분야에서 실제 소프트웨어를 개발하는 사례 중심으로 이루어지고 있으며, 바이브 코딩이 특정한 개발 환경에 한정된 기술이 아니라 다양한 도메인의 소프트웨어 개발에 적용될 수 있는지를 탐색하는 단계에 있다.

결론적으로 바이브 코딩은 EUC, PBD, LCNC에 이르기까지 지속되어 온 프로그래밍 추상화의 확대와 구현 세부 사항의 위임이 LLM을 통해 새로운 단계에 도달한 현상으로 해석할 수 있다. 따라서 단순히 비프로그래머를 위한 새로운 프로그래밍 방식으로 바이브 코딩을 이해하기보다, 프로그래밍 작업을 인간과 컴퓨팅 시스템 사이에 어떻게 분배할 것인가라는 보다 광범위한 변화 안에서 이를 바라볼 필요가 있다. 이러한 관점에서 바이브 코딩은 프로그래밍 접근성 확대와 전문 개발자 역량 증강을 동시에 포함한다. 그러나 바이브 코딩에 관한 기존 연구는 아직 초기 단계에 있으며, 개념적 논의와 특정 도메인에서의 적용 사례가 개별적으로 축적되고 있어 바이브 코딩 연구가 어떠한 주제와 관점으로 구성되고 있는지에 대한 전체적인 연구

www.kci.go.kr

지형은 충분히 파악되지 않았다. 특히 바이브 코딩이 비프로그래머의 프로그래밍 접근성을 확대할 뿐만 아니라 전문 개발자의 개발 방식과 프로그래밍 작업의 분배를 변화시키는 연구와도 연결된다는 점을 고려하면, 개별 연구의 내용을 검토하는 것을 넘어 기존 연구들이 어떠한 주제를 중심으로 연결되고 발전했는지 종합적으로 확인할 필요가 있다.

3. Existing literature review studies on user-level programming abstraction

앞서 살펴본 바와 같이 최종 사용자 수준 프로그래밍 추상화의 다양한 접근이 제안되었으며, 이러한 연구 흐름은 개별 기술이나 방법론에 관한 연구뿐 아니라 특정 영역에 축적된 문헌을 체계적으로 종합하는 문헌 분석 연구로도 확장되었다. 예컨대 Barricelli et al.[37]은 최종 사용자 개발(End-user development; 이하 EUD), 최종 사용자 프로그래밍(End-user programming; 이하 EUP), 최종 사용자 소프트웨어 공학(End-user software engineering; 이하 EUSE) 연구를 대상으로 2000년부터 2017년까지의 165편의 논문을 분석했다. 분석 대상 문헌들은 접근법의 유형, 상호작용 기법, 개발 과정의 단계, 응용 분야, 사용자 유형, 활용 목적 및 평가 방법 등의 차원으로 분류되었으며, EUD·EUP·EUSE 연구가 다양한 분야에서 지속적으로 확장되었음을 확인했다. 그러나 각 연구 영역 사이의 연계와 지식 교류는 제한적이었으며, 서로 다른 접근을 통합적으로 설명할 수 있는 공통적인 연구 틀도 충분히 형성되지 않은 것으로 나타났다. Santos & Villela[38] 역시 EUD 솔루션을 대상으로 체계적 문헌 분석을 수행해 선행 연구들이 주로 웹 기반 환경, 기존 애플리케이션의 수정·확장, 시각적 프로그래밍 및 구성 기반의 상호작용을 활용함을 보고했다. Ko et al.[39]은 EUSE의 연구 동향을 종합적으로 검토하고, 사용자가 단순히 프로그램을 작성하는 것뿐만 아니라 요구사항 정의, 명세, 재사용, 통합, 테스트, 검증 및 디버깅과 같은 소프트웨어 공학의 다양한 활동에서 어려움을 경험한다는 점을 정리했다.

PBD 분야에서의 문헌 분석 연구는 많지 않지만, PBD와 개념적·방법론적으로 밀접하게 관련된 시연을 통한 학습(Learning from Demonstration; 이하 LfD) 관련 문헌 분석은 일부 연구에서 이루어졌다. Argall et al.[40]은 로봇 학습의 LfD 분야에서 사용자의 시연을 어떻게 획득하고 표현하며 이를 새로운 행동이나 정책으로 어떻게 일반화하는지를 중심으로 관련 연구를 체계적으로 정리했다. Ajaykumar et al.[41]은 최종 사용자 로봇 프로그래밍(End-user robot programming)에 관한 연구를 종합하여 PBD를 포함한 다양한 프로그래밍 접근법을 사용자의 프로그램 명세 방식, 프로그래밍 과정에서의 사용자 참여, 상호작용 방식 및 평가 방법의 관점에서 비교했다. 해당 연구는 사용자가 프로그램의 내부 구조를 직접 기술하는 방식뿐만 아니라 시연, 자연스러운 상호작용 및 고수준 명세를 이용하는 방식이 함께 발전해 왔음을 보여주었다.

LCNC는 문헌 분석 연구가 비교적 활발하게 이루어진 분야이다. Bucaioni et al.[42]은 LC 개발 관련 58편의 선행 문헌을 체계적으로 분석했다. 분석 결과 LC 개발 연구는 빠르게 증가하고 있었으며, 모델 주도 공학(Model-driven engineering)과의 연계가 중요한 기술적 기반으로 나타났다. 또한 추상화와 자동화를 통해 개발 과정을 단순화하고 개발 생산성을 높이는 것이 주요한 기대효과로 나타났으며, 학술 문헌뿐만 아니라 산업 현장의 자료를 함께 고려할 필요성이 제기되었다. Binzer & Winkler[43]는 LCNC의 연장선에서 시민 개발(Citizen development) 관련 체계적 문헌 분석을 수행해 79개의 학술 및 실무 자료를 검토하고, 시민 개발의 정의와 특성, 도입의 촉진 요인과 장애요인, 조직적 도입 전략, 역량 등을 주요 연구 주제로 도출했다. 이는 LCNC가 단순히 프로그래밍을 쉽게 만드는 기술을 넘어 소프트웨어 개발에 참여하는 사용자와 조직 내 개발 역할을 변화시키는 현상으로 연구되었음을 보여준다. Ajimati et al.[44]의 체계적 문헌 분석 연구에서는 2017-2023년의 40개 논문을 분석해 LCNC 연구가 시민 개발, 조직적 도입, 디지털 전환, 생산성 및 개발 효율성뿐만 아니라 보안, 거버넌스, 기술적 한계와 같은 문제까지 다루고 있음을 확인했다.

바이브 코딩에 관한 문헌 분석 연구는 앞선 연구 영역에 비해 제한적이나, 일부 연구는 실무 자료와 같은 회색 문헌(Grey literature)을 포함해 문헌 분석을 수행했다. Fawzy et al.[12]은 바이브 코딩의 실제 활용 경험을 다룬 회색 문헌을 대상으로 101개의 자료에서 518개의 실제 행동 사례를 분석했다. 연구 결과 빠른 개발 속도와 접근성이 주요 활용 동기로 나타났으나, 생성된 코드의 품질과 검증에 관한 문제가 반복적으로 보고되었으며, 테스트를 생략하거나 생성 결과의 검증을 인공지능에 다시 의존하는 사례도 확인되었다. Siddeeq et al.[3]은 학술 문헌과 회색 문헌을 함께 분석하는 다중 자료 문헌 고찰(Multivocal literature review)을 수행해 28편의 학술 문헌과 19편의 회색 문헌을 분석했다. 분석 결과 바이브 코딩은 일회적인 코드 생성보다는 생성, 평가, 수정이 반복되는 과정으로 나타났으며, 개발자의 역할 역시 직접적인

www.kci.go.kr

코드 작성에서 요구사항 명세, 감독 및 검증으로 이동하는 경향이 확인되었다. 또한 생산성 및 프로토타이핑 시간 단축에 관한 긍정적인 결과가 보고된 반면, 장기적인 유지보수성, 소프트웨어 품질 및 안전성에 관한 연구 근거는 상대적으로 제한적인 것으로 나타났다.

이상의 연구들은 사용자 수준 프로그래밍 추상화의 다양한 영역에서 연구 주제, 사용자 참여 방식, 상호작용 방식, 개발 과정 및 평가 방법 등이 체계적으로 축적되었음을 보여준다. 최근 바이브 코딩 관련 문헌 분석 연구에서도 생산성 및 접근성 향상과 함께 코드 품질, 검증, 유지보수 및 안전성 등의 한계가 주요 쟁점으로 나타남을 확인했다[3]. 그러나 기존의 바이브 코딩 문헌 분석 연구는 회색 문헌이나 다중 자료를 중심으로 수행되었으며, 국내외 학술 연구를 대상으로 전반적인 양상과 주제 등을 종합적으로 비교·분석한 연구는 찾아보기 어렵다.

III. The Proposed Scheme

본 연구는 바이브 코딩과 관련된 국내외 학술 연구 동향을 비교·분석하고자 데이터 수집 및 전처리, TF-IDF 기반 키워드의 시계열 분석, 토픽 모델링의 과정으로 연구를 진행했다.

1. Data collection and preprocessing

바이브 코딩 관련 학술 문헌은 Web of Science(이하 WoS)와 한국학술지인용색인(Korea Citation Index; 이하 KCI)에서 수집했다. WoS와 KCI는 과학, 사회과학, 인문학 등 다양한 학문 분야에서 신뢰도 높은 인용 색인 데이터를 제공하기 때문에 문헌 분석 연구에 널리 활용된다는 점을 고려하여 검색 대상 데이터베이스로 선정했다[45, 46]. 여기서 국내외 문헌의 분류는 연구자의 국적이나 연구 수행 국가를 기준으로 한 것이 아니라 학술지의 국내외 출판·색인 범위를 기준으로 한 것이다. 따라서 본 연구에서 '국제 연구'와 '국내 연구'는 각각 WoS 및 KCI에 수록된 문헌을 뜻한다.

데이터 수집은 2026년 8월 4일에 수행했으며, 검색어는 국내외 데이터베이스 모두 'vibe coding'을 사용했다. 별도의 검색어 필드나 유사어 설정은 하지 않았다. 검색 대상 문헌의 출판 시기는 바이브 코딩이라는 용어가 등장한 2025년 이후로 설정했으며, 수집된 문헌으로부터 저자명, 출판연도, 저널명, 권·호·페이지 정보, 초록, 키워드 등의 서지 정보를 확보했다. 또한 서지 정보를 바탕으로 중복 여부를 검토해 검색 결과에 중복 문헌이 없음을 확인했다.

분석 대상 문헌 선정 기준으로 (1) 저널 논문일 것, (2) 내용 분석에 필요한 영문 초록이 제공될 것, (3) 제목 및 초록을 기준으로 바이브 코딩과 직접적인 관련성이 확인될 것 등의 세 기준을 적용했다. 여기서 기준 (3)은 논문의 제목 또는 초록에 'vibe'와 'coding'의 단어가 포함되어 검색 결과로 추출되었으나, 해당 단어가 본 연구 주제인 바이브 코딩과 무관한 다른 연구 대상이나 도메인을 다루는 문헌은 제외했음을 의미한다.

WoS에서는 총 72건의 문헌이 검색되었다. 이 중 학술 연구의 동향 분석이라는 본 연구 목적에 부합하도록 저널 논문만을 분석 대상으로 선정했으며, 이에 따라 저널 논문에 해당하지 않는 16건의 문헌이 제외되었다. 이후 수집된 문헌의 제목 및 초록을 검토해 바이브 코딩과의 관련성이 확인되지 않은 문헌 및 저널 규정상 초록이 제공되지 않아 내용 분석이 어려운 문헌 17건을 추가로 제외했다. 이러한 과정을 통해 WoS에서는 최종적으로 39건의 문헌을 분석 대상으로 선정했다.

KCI에서는 검색 결과가 모두 저널 논문에 해당했으므로 별도의 유형 필터링 없이 46건의 문헌을 1차 확보했다. 이후 문헌의 제목 및 초록을 검토해 바이브 코딩과 직접적인 관련성이 없는 2건의 문헌을 제외해 최종적으로 44건의 국내 문헌을 분석 대상으로 선정했다. 이상의 과정을 거쳐 수집된 분석 대상 문헌은 총 83건이었다.

Fig. 1은 분석 대상 논문의 월별 출판 수를 국제 연구와 국내 연구로 구분해 나타낸 것이다.

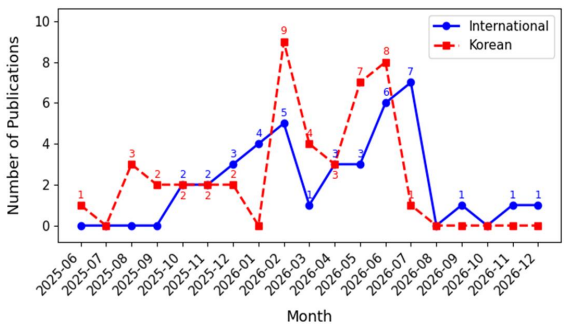

Fig. 1. Monthly Distribution of Publications on Vibe Coding

국내에서는 2025년 6월에, 해외에서는 2025년 10월에 바이브 코딩 관련 연구가 발표되기 시작했으며, 2026년으로 접어들면서 전반적으로 연구가 증가하는 양상을 보였다. 특히 국제 연구는 2026년 2월과 6-7월에, 국내 연구는 2026년 2월과 5-6월에 다수의 논문이 발표되었다. 2026년 8월 이후로 표시되는 일부 문헌은 정식 권·호에 게

www.kci.go.kr

재되기 이전의 Online First 논문을 포함한다. 이를 고려할 때 바이브 코딩에 대한 학술적 관심과 연구 활동은 2025년 이후 점차 확대되는 추세로 해석할 수 있다.

2. Temporal changes in TF-IDF-based keywords

바이브 코딩 관련 연구의 시기별 주요 연구 주제와 관심 영역의 변화를 파악하기 위해 용어 빈도-역문서 빈도(Term frequency-Inverse document frequency; 이하 TF-IDF) 기반 키워드 시계열 분석을 수행했다. 데이터 수집 단계에서 각 논문의 저자 키워드도 확보하긴 했으나, 저자 키워드는 연구자가 임의로 선정할 수 있어 문헌에 나타난 주요 개념을 일관되게 파악하는 데 한계가 있을 수 있으므로[45], 본 연구에서는 연구 내용을 더 직접적으로 반영하고자 저자 키워드 대신 초록을 대상으로 TF-IDF 분석을 적용했다. 또한 국내외 비교의 용이성을 위해 모든 문헌의 영문 초록을 분석했다.

텍스트 전처리를 위해 전체 텍스트를 소문자로 변환하고 구두점을 제거했으며, sklearn.feature_extraction.text 모듈로 불용어(Stopwords)를 제거했다. 아울러 바이브 코딩 연구에서 빈번하게 등장하지만 연구 주제 간 차이를 구분하는 데 정보성이 낮은 'vibe', 'coding', 'vibecoding', 'AI', 'LLM', 'using', 'used', 'paper', 'study', 'based' 등의 단어를 추가적인 사용자 정의 불용어로 지정했다. 또한 전처리 과정에서 두 글자 이하의 토큰은 제외했다.

전처리된 영문 초록 텍스트는 연구 시기에 따른 주제 변화를 분석하기 위해 분기 단위로 구분되었다. 다만 바이브 코딩 관련 연구가 2025년 이후 본격적으로 등장했고, 특정 시기에는 문헌 수가 매우 적다는 특성을 고려해 문헌 수가 상대적으로 적은 인접 분기를 통합했다. 국제 문헌은 2025년 4분기, 2026년 1분기, 2026년 2분기, 2026년 3-4분기의 네 시기로 구분했으며, 국내 문헌은 2025년 2-3분기, 2025년 4분기, 2026년 1분기, 2026년 2-3분기의 네 시기로 구분했다. 본 연구에서 분류한 국내외 문헌의 분기별 출판 수는 Fig. 2에 제시했다.

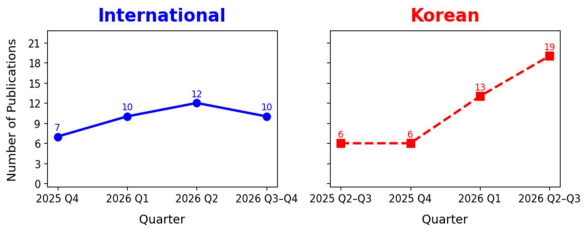

Fig. 2. Quarterly Distribution of Publications on Vibe Coding

각 시기에 포함된 문헌의 초록을 통합하여 코퍼스(Corpus)를 구축하고, 각 시기의 통합 코퍼스를 하나의 문서(Document)로 간주해 이를 대상으로 TF-IDF 분석을 수행했다. 따라서 국제 및 국내 문헌 각각에서 4개의 시기별 문서가 구성되며, IDF는 각 집단 내 4개의 시기별 문서를 기준으로 산출되었다.

TF-IDF는 특정 문서 또는 문서 집합에서 특정 단어가 갖는 상대적인 중요도를 평가하는 방법으로, 단순한 출현 빈도뿐만 아니라 전체 문서에서의 희소성을 함께 고려할 수 있다는 장점이 있다. 본 연구에서는 국제 문헌과 국내 문헌에 대해 각각 별도의 TF-IDF 분석을 수행했으며, 그 과정은 다음과 같다. 먼저 scikit-learn의 TfidfVectorizer를 사용해 각 집단에서 전체 분석 기간에 걸쳐 상위 15개의 주요 단어를 선정했다. 단어의 최소·최대 출현 빈도에 대한 별도의 제한은 두지 않았으며, 단어 단위(Unigram)를 사용하고 기본 TF-IDF 설정인 IDF 적용 및 L2 정규화를 사용했다. 이에 따라 각 집단에서 선정된 15개 주요 단어에 대해 시기별 TF-IDF 값을 산출함으로써 각 시기의 통합 코퍼스에서 상대적으로 특징적인 어휘를 중심으로 연구 주제의 변화를 탐색적으로 파악했다. 또한 국제 문헌과 국내 문헌은 서로 독립적으로 TF-IDF를 산출했으므로, 두 집단의 TF-IDF 값의 절대적 크기를 직접 비교하기보다는 각 집단 내부에서 시기에 따른 상대적 변화 양상을 중심으로 해석했다. 국내외 문헌에서 도출된 시기별 TF-IDF 값은 히트맵으로 시각화했다.

각 시기의 연구 내용을 대표하는 문헌을 파악하기 위해 코사인 유사도(Cosine similarity)를 활용했다. 시기별 TF-IDF 벡터를 해당 시기의 대표적인 주제적 특성을 나타내는 벡터로 설정하고, 해당 시기에 포함된 개별 논문의 초록 벡터와 코사인 유사도를 계산했다. 이후 유사도가 높은 순으로 논문을 정렬해 각 시기의 대표 논문을 선정했다. 이를 통해 시기별 키워드와 함께 해당 시기의 연구 주제를 구체적으로 보여주는 대표 연구를 함께 검토했다.

Fig. 3는 국제 문헌에서 분기별 초록의 TF-IDF 값을 분석한 결과이다. 2025년 4분기에는 development의 TF-IDF 값이 가장 컸으며, clinical, tools, learning, code 등도 주요 키워드로 나타났다. 이는 바이브 코딩 관련 초기 연구가 소프트웨어 개발(development) 자체에 관한 논의와 함께 특정 분야에서의 AI 기반 도구 개발 및 활용 가능성을 탐색하는 방향으로 이루어졌음을 보여준다. 해당 시기의 대표 논문[4]은 신장학 교육을 대상으로 AI 지원 방식의 대화형 교육 도구를 개발하고, 임상 전문가의

www.kci.go.kr

지식과 추론 과정을 자연어 기반 코드 생성 및 반복적 개선 과정에 반영하는 연구였다. 이는 초기 바이브 코딩 연구가 개발을 중심으로 의학 및 교육과 같은 구체적인 응용 분야와 결합해 나타났음을 보여준다.

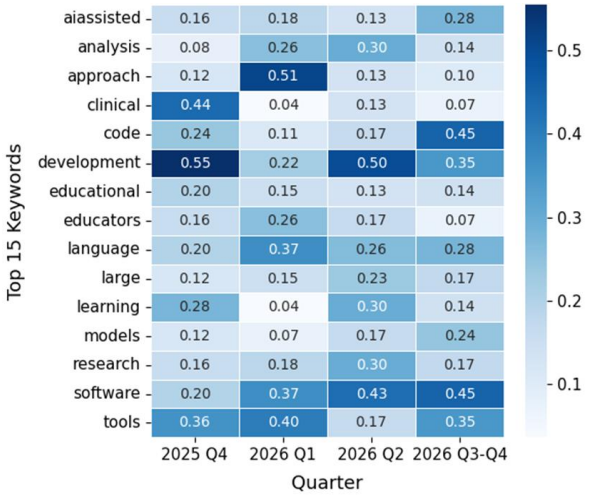

Fig. 3. Temporal Changes in TF-IDF-Based Keywords in International Vibe Coding Research

2026년 1분기에는 approach를 비롯해 tools, language, software 등이 주요 키워드로 확인되었다. 이는 바이브 코딩에 관한 연구가 특정 응용 사례를 넘어 하나의 개발 접근법(approach)으로 개념화되기 시작했음을 시사한다. 특히 language와 software가 주요 키워드로 나타난 것은 자연어를 통한 개발 의도 전달과 소프트웨어 생성 간의 관계가 주요 연구 관심사로 부상했음을 보여준다. 해당 시기의 대표 논문[47]에서는 바이브 코딩을 '자연어로 개발 의도를 표현하고 생성형 AI 및 LLM을 통해 소프트웨어 산출물을 생성하는 사회·기술적 프레임워크'로 정의하고, 신뢰성, 보안, 거버넌스 및 인간 중심의 적응 등을 주요 연구 과제로 제시하였다. 종합하면 이 시기에는 바이브 코딩의 개념적 정립과 개발 접근법으로서의 가능성에 대한 논의가 확대된 것으로 볼 수 있다.

2026년 2분기에는 development와 software가 높은 TF-IDF 값을 나타냈으며, analysis, learning, research 등이 뒤를 이었다. 이는 앞선 시기에 나타난 바이브 코딩의 개념적·방법론적 논의가 실제 소프트웨어 개발 및 연구 과정으로 구체화하는 양상을 보여준다. 특히 development와 software의 높은 중요도는 바이브 코딩을 실제 개발 과정에 적용하고 그 효과와 가능성을 검토하는 연구가 지속적으로 이루어졌음을 시사한다. 해당 시기의 대표 논문[48]은 LCNC 플랫폼에 대한 체계적 문헌 고찰을 통해 소프트웨어 개발의 민주화와 의도 기반 개발로의 변화를 분석했으며, 이러한 변화의 주요 동향으로 바이브 코딩을 제시했다. 이는 바이브 코딩이 기존의 LCNC 개발 환경과 함께 프로그래밍 지식의 요구 수준을 낮추고 사용자의 개발 의도를 직접적으로 소프트웨어 구현으로 연결하는 새로운 개발 패러다임의 일부로 논의되었음을 보여준다.

2026년 3-4분기에는 code와 software가 가장 높은 TF-IDF 값을 나타냈으며, development, tools, language, models 등도 주요 키워드로 확인되었다. 특히 code와 software의 중요도가 높아진 것은 바이브 코딩 연구가 초기의 개념적 논의를 넘어 실제 코드 생성과 소프트웨어 구현 및 자동화에 관한 기술적 연구로 점차 구체화함을 시사한다. 해당 시기의 대표 논문[49]은 다양한 개발 산출물(사용 사례, UI 설계, API 계약, 기존 코드 베이스 등)을 활용해 웹사이트 기능을 자동으로 생성하는 바이브 코딩 기반 방법을 제안했다. 이는 단순한 자연어 기반 코드 생성에 그치지 않고, 다양한 개발 정보를 통합하여 소프트웨어 개발 과정을 자동화하는 방법론으로 바이브 코딩을 확장하려는 연구 방향을 보여준다.

종합하면, 국제 바이브 코딩 연구는 2025년 4분기의 개발 및 특정 분야 응용에서 출발해 2026년 1분기의 개발 접근법과 자연어 기반 상호작용에 관한 개념적 논의를 거친 후, 2026년 2분기 이후 소프트웨어 개발과 코드 생성 및 자동화를 중심으로 구체화하는 양상을 보였다. 즉 국제 연구에서는 바이브 코딩의 활용 가능성을 탐색하는 초기 연구에서 점차 실제 소프트웨어 개발 프로세스에 적용하고 자동화하는 방향으로 연구의 초점이 이동하고 있는 것으로 해석할 수 있다.

국내 문헌을 대상으로 분기별 초록의 TF-IDF 값을 분석한 결과, 국내 바이브 코딩 연구는 국제 연구와 비교하여 교육 및 학습 분야를 중심으로 연구가 전개되는 특징이 뚜렷하게 나타났다(Fig. 4 참조). 2025년 2-3분기에는 educational과 language를 포함해 development, education, tools 등이 주요 키워드로 확인되었다. 이는 국내에서 바이브 코딩 관련 연구가 등장한 초기부터 교육 분야를 중심으로 AI 기반 도구를 개발하고 활용하는 방향으로 이루어졌음을 보여준다. 해당 시기의 대표 연구[6]는 바이브 코딩을 활용해 중국어 학습을 위한 게임형 웹 애플리케이션을 개발했다. 구체적으로 Replit과 같은 클라우드 기반 개발 환경과 ChatGPT, Claude, Gemini 등의 LLM을 활용하여 문장 배열, 어휘 플래시 카드, 타이핑 게임, 성조 매칭, AI 챗봇 역할극 등 다양한 중국어 학습용 웹 애플리케이션을 구현했다. 이러한 결과는 초기 국내 연구에서 바이브 코딩이 교육 콘텐츠 개발을 위한 도구로 활용

되었으며, 특히 비프로그래머 또는 교육자가 AI를 활용해 직접 학습 도구를 제작할 수 있다는 가능성에 주목했음을 보여준다.

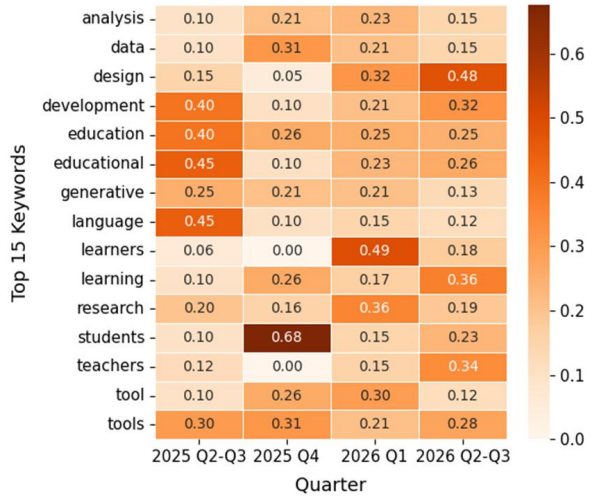

Fig. 4. Temporal Changes in TF-IDF-Based Keywords in Korean Vibe Coding Research

2025년 4분기에는 students가 가장 높은 TF-IDF 값을 나타냈으며, educational, data, tools, learning, education 등도 주요 키워드로 나타났다. students의 높은 중요도는 바이브 코딩 연구의 관심이 교육용 도구의 개발 자체에서 학생의 교육 경험 및 학습 성과로 확장되고 있음을 시사한다. 해당 시기의 대표 논문[10]은 컴퓨터 공학 비전공 대학생을 대상으로 생성형 AI, 데이터 분석, 업무 자동화, NC 및 바이브 코딩을 포함한 실습 중심 교육 과정을 운영하고 학습 결과물을 분석했다. 이를 통해 학생들이 바이브 코딩과 NC 도구를 활용해 웹사이트, 온라인 전시, AI 기반 서비스 애플리케이션 등 다양한 프로젝트를 제작할 수 있음을 확인했다. 따라서 이 시기에는 바이브 코딩이 단순한 교육용 콘텐츠 제작 도구를 넘어 비전공 학습자의 실무 역량 및 소프트웨어 개발 경험을 지원하는 교육 방법으로 논의되기 시작한 것으로 해석할 수 있다.

2026년 1분기에는 learners가 가장 높은 TF-IDF 값을 보였으며, research, design, tool, analysis 등이 그 뒤를 이었다. 이는 바이브 코딩을 활용한 교육적 실천과 학습자 경험을 체계적으로 분석하고 설계하는 연구가 증가한 것으로 볼 수 있다. 해당 시기의 대표 논문[8]은 LLM 기반 플랫폼을 활용한 진로상담 챗봇 설계 실습에 바이브 코딩 접근법을 적용하고, 참여적 실행연구를 통해 학습자 경험을 분석했다. 연구 결과 학습자들은 별도의 코딩 지식 없이 AI와의 대화를 통해 챗봇을 직접 설계·구현하면서 개발에 대한 성취감을 경험했으며, 프롬프트 설계와 페르소나 구성 등 AI 활용 역량과 상담 전문성이 결합하는 과정을 경험했다. 이는 국내 바이브 코딩 연구가 교육 도구의 개발에서 학습자의 실제 경험과 역량 형성에 대한 연구로 확장됨을 보여주었다.

2026년 2-3분기에는 design을 비롯해 learning, teachers, development, education 등이 주요 키워드로 확인되었다. teachers와 design의 중요도가 높아진 것은 바이브 코딩의 주체가 학습자뿐만 아니라 교사와 교육 실천가로 확대되고 있으며, 교육적 목적에 맞는 디지털 학습 도구를 직접 설계하고 개발하는 과정이 연구 관심사로 부상하고 있음을 시사한다. 해당 시기의 대표 논문[11]은 초등교사가 ChatGPT와 Canva를 활용해 바이브 코딩 방식으로 음악 학습용 웹 애플리케이션을 직접 설계·개발하는 과정을 분석했다. 해당 연구에서는 교사가 별도의 전문적인 코딩 지식 없이 5개의 게임 기반 음악 학습 웹 애플리케이션을 개발했으며, 전문가 검토를 통해 교육적 의도가 디지털 학습 도구의 기능과 상호작용에 반영될 수 있음을 확인했다. 이는 국내 바이브 코딩 연구가 학습자 대상 연구를 넘어 교사가 직접 교육용 소프트웨어를 설계·개발하는 방향으로 확장된 사례이다.

결론적으로 국내 바이브 코딩 연구는 2025년 2-3분기의 교육 콘텐츠 및 학습 도구 개발에서 출발해 2025년 4분기에는 학생의 프로젝트 수행과 실무 역량으로 관심이 확대되었으며, 2026년에는 학습자 경험과 교사의 교육적 설계 및 개발 활동으로 연구 대상과 범위가 확장되는 양상을 보였다. 특히 국내 연구에서는 전 기간에 걸쳐 교육 관련 키워드가 지속적으로 나타나고 있어, 바이브 코딩이 주로 교육 현장에서 프로그래밍 비전문가가 LLM을 활용해 직접 디지털 학습 도구를 설계·개발하고 학습자 및 교사의 역량을 강화하는 수단으로 연구되고 있음을 확인했다. 이는 앞서 확인한 국제 연구가 소프트웨어 개발 및 자동화로 연구 관심을 확장하는 것과 비교되는 국내 연구의 특징으로 볼 수 있다.

3. Topic modeling

바이브 코딩 관련 연구에서 나타나는 주요 연구 주제를 체계적으로 도출하고 주제 간 구조적 관계를 파악하기 위해 잠재 디리클레 할당(Latent Dirichlet Allocation; 이하 LDA) 기반의 토픽 모델링을 수행했다. 토픽 모델링은 문헌에 포함된 단어들의 동시 출현 패턴을 바탕으로 문헌 집합에 잠재적으로 존재하는 주제를 추출하는 방법으로, 본 연구에서는 토픽 모델링을 문헌 분석에 사용한 선행 연구를 고려하여 각 논문의 영문 초록을 분석 단위로 설정했

다[45, 46]. 특히 바이브 코딩 관련 학술 연구가 형성되는 초기 단계에 나타난 주요 주제를 탐색적으로 파악하고자 하는 본 연구의 목적을 고려해, 문헌별 주제 분포 및 토픽을 구성하는 주요 단어를 해석할 수 있는 표준 LDA를 적용했다. 본 연구의 분석 대상 문헌이 바이브 코딩이라는 용어가 등장한 이후의 제한된 기간에 수집된 비교적 적은 수의 문헌으로 구성되어 있어, 시간과 문헌 출처를 동시에 모형에 포함하는 확장 모형보다는 문헌에 내재된 잠재 주제의 구조를 탐색하는 표준 LDA가 적합하다고 보았다. 다만 표준 LDA는 문헌의 메타데이터를 모형 내에서 직접 설명하거나 시간에 따른 토픽 변화 및 집단 간 토픽 비중의 차이를 추정하는 데에는 한계가 있으므로, 시간적 변화는 앞서 수행한 키워드 시계열 분석을 병행해 해석했다.

초록 텍스트 전처리는 키워드 시계열 분석과 같은 방식으로 수행했으며, 이후 문헌별 토큰(Token)을 생성했다. 전처리된 토큰을 기반으로 단어-문헌 사전(Dictionary)과 문헌-단어 행렬(Corpus)을 구축한 뒤 LDA 모델을 적용했다. 적절한 토픽 수 결정을 위해 토픽 수 K를 2개에서 9개까지 변화시키면서 각 모형의 퍼플렉시티(Perplexity)와 코히어런스(Coherence)를 산출했다. 퍼플렉시티는 확률적 언어 모델이 주어진 문헌을 얼마나 잘 설명하는지를 나타내는 지표로, 일반적으로 값이 낮을수록 모델의 예측 성능이 우수한 것으로 해석된다. 반면 코히어런스는 하나의 토픽을 구성하는 주요 단어들이 의미적으로 얼마나 일관된 관계를 가지는지를 평가하는 지표로, 값이 높을수록 토픽의 해석 가능성과 의미적 일관성이 높은 것으로 판단할 수 있다. 본 연구에서는 단일 지표만을 기준으로 토픽 수를 결정하지 않고, 선행 연구들[45, 46]에서 활용한 퍼플렉시티와 코히어런스를 종합적으로 고려해 최적의 토픽 수를 결정했다. 국제와 국내 문헌에서 토픽 수에 따른 퍼플렉시티와 코히어런스 값은 Fig. 5에 제시했다. 그림에서 빨간색 실선은 퍼플렉시티, 파란색 점선은 코히어런스 값이다.

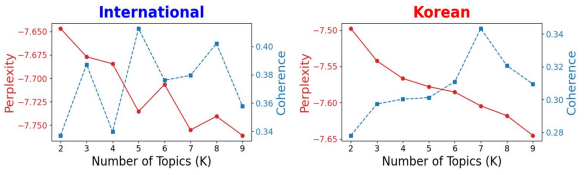

Fig. 5. Perplexity and Coherence Scores by Number of Topics for International and Korean Literature

국제 문헌은 토픽 수 K=5일 때, 국내 문헌은 K=7일 때 코히어런스가 최대가 되었으며, 해당 K에서 퍼플렉시티의 감소 경향도 완화되었다. 따라서 국제 문헌에서는 5개의 토픽, 국내 문헌에서는 7개의 토픽을 사용했다. 이후 각 토픽을 구성하는 단어 중 토픽 내 가중치가 높은 상위 10개 단어를 추출해 토픽의 주요 내용을 파악했다. 토픽 간의 관계와 분포는 pyLDAvis 라이브러리로 시각화했다.

각 토픽의 구체적인 연구 내용을 확인하고자 토픽별 대표 논문을 1편씩 선정했다. 분석 대상 문헌에 대해 LDA 모델이 산출한 토픽별 확률(Topic probability)을 계산한 후, 특정 토픽에 대한 확률이 가장 높은 문헌을 해당 토픽의 대표 논문으로 사용했다. 즉, 각 토픽에 가장 높은 확률로 할당된 문헌을 해당 토픽의 연구 특성을 가장 잘 나타내는 대표 문헌으로 간주했다. 마지막으로 토픽별 상위 10개 단어와 대표 논문을 종합하여 LLM을 보조 도구로 사용해 각 토픽의 이름을 명명했다. 토픽 명칭 지정에 LLM을 활용하는 것은 토픽별 데이터를 바탕으로 명칭 후보군을 도출해 명명의 체계성을 확보하고자 최근 연구에서 사용되고 있는 방법이다[45, 46]. 다만 LLM을 이용한 토픽 명명은 연구자의 해석을 제거하는 절차가 아니라, 토픽 명칭 후보군을 생성하는 과정의 일부를 LLM에 위임하는 것으로 보았다. 따라서 LLM이 제시한 명칭을 최종 결과로 직접 사용하지 않고, 본 연구의 저자가 토픽별 상위 단어와 대표 논문의 내용이 후보 명칭에 적절하게 반영되었는지를 검토한 후 최종 명칭을 결정했다.

LLM에 제공한 프롬프트는 (1) 본 연구의 전체 맥락, (2) 각 토픽의 상위 10개 단어, (3) 각 토픽의 대표 논문 영문 초록으로 구성되었다. 토픽별 명칭에 각 토픽이 다른 토픽과 구분되는 특징이 드러날 수 있도록, 개별 토픽 데이터를 별도의 프롬프트가 아닌 하나의 프롬프트에 포함하여 토픽별 비교가 이루어지도록 했다. 사용된 LLM은 GPT-5.6 Luna였다. 본 연구의 토픽 모델링 시점에 발표된 가장 최신 LLM은 OpenAI의 GPT-5.6 계열이었으며, 토픽 명칭 도출 과정에서 전체 연구 맥락, 토픽별 단어, 대표 논문 내용 등 여러 요소를 포함하는 프롬프트 특성상 대규모 토큰의 컨텍스트 윈도우를 제공하는 GPT-5.6 Luna가 적합하다고 판단했다. LLM에는 제공된 정보에 근거해 토픽의 핵심 연구 대상과 방법을 간결하게 나타내는 토픽 명칭 후보를 제안하도록 요청했으며, 제공하지 않은 새로운 연구 주제나 내용을 임의로 추가하지 않도록 지시했다. 생성된 후보 명칭에 대해 본 연구의 저자가 해당 토픽의 상위 단어와 대표 논문 초록을 다시 검토하여 의미적 일치 여부와 다른 토픽과의 구별 가능성을 확인하고, 필요한 경우 명칭을 수정했다.

Fig. 6는 국제 연구의 토픽 모델링 결과 도출된 5개 토픽의 거리 지도(Intertopic distance map)이고, Table 1은 각 토픽에서 상위 10개 단어, 토픽 명칭, 대표 논문을 정리한 표이다.

www.kci.go.kr

Table 1. Topics Derived from International Literature Related to Vibe Coding

Topic	Top 10 Most Relevant Terms	Topic Name	Representative Literature
1	tools, learning, code, research, clinicians, available, cli, medical, generative, gemini	Clinical Research and Vibe Coding Tools	Lee & Huh [32]
2	approach, software, development, bug, research, language, llms, learning, design, intelligence	AI-Assisted Software Development Approach	Otsuki & Nuri [50]
3	development, code, aiassisted, educational, learning, interface, digital, use, education, tools	AI-Assisted Educational Tool Development	Rahman [7]
4	software, tools, analysis, language, words, governance, large, research, generative, development	Responsible Software Development and Governance	García [51]
5	development, clinical, aiassisted, tools, information, language, code, engineering, models, software	AI-Assisted Clinical Software Development	Kıyak et al. [5]

Fig. 6. Intertopic Distance Map of International Literature Related to Vibe Coding

먼저 토픽 1은 '임상 연구와 바이브 코딩 도구(Clinical research and vibe coding tools)'로 명명했다. 주요 단어로 tools, clinicians, medical, code 등이 나타났으며, 대표 논문[32]은 임상의가 머신러닝 및 딥러닝 연구를 수행하는 과정에서 ChatGPT, Gemini, Claude, Replit 및 CLI 기반 도구의 활용 방법을 제시했다.

토픽 2는 'AI 기반 소프트웨어 개발 접근법(AI-assisted software development approach)'으로 명명했다. approach, software, development, bug 등이 주요 단어로 나타난 이 토픽의 대표 논문[50]은 바이브 코딩을 활용해 의학적 진단 목적의 3차원 형태 분석 소프트웨어를 직접 개발한 사례를 보여주었다.

토픽 3은 'AI 기반 교육 도구 개발(AI-assisted educational tool development)'로 명명했다. 주요 단어는 educational, learning, education, tools 등이었고, 대표 논문[7]은 간호교육에서 바이브 코딩을 적용해 교육자와 학습자가 디지털 교육 도구의 단순한 사용자를 넘어 직접적인 공동 설계자 및 제작자로 참여할 가능성을 논의하고 정확성, 편향, 개인정보 보호 및 교육적 품질 등의 잠재적 문제를 제기했다.

토픽 4는 '책임 있는 소프트웨어 개발과 거버넌스(Responsible software development and governance)'로 명명했다. 관련 단어 중 governance, generative, software가 두드러졌으며, 대표 논문[51]은 의학 교육에서 바이브 코딩을 교수 역량 관점에서 검토할 때 생성된 결과물의 정확성·안전성을 확보하기 위해서는 해당 분야 전문 지식이 중요한 검증 수단으로 작용한다는 점을 강조했다.

토픽 5는 'AI 기반 임상 소프트웨어 개발(AI-assisted clinical software development)'로 명명했다. 주요 단어는 clinical, development, tools, code, software 등이었고, 대표 논문[5]은 바이브 코딩으로 개발한 웹 기반 시스템을 활용해 의과대학생의 외과적 진단 추론을 지원하는 자동화 학습 시스템을 설계하는 연구였다.

국내 연구의 토픽 모델링 결과 도출된 7개 토픽에서 거리 지도와 세부 정보는 Fig. 7 및 Table 2에 제시했다.

www.kci.go.kr

Table 2. Topics Derived from Korean Literature Related to Vibe Coding

Topic	Top 10 Most Relevant Terms	Topic Name	Representative Literature
1	learning, web, tools, music, elementary, learners, technical, app, support, design	Vibe Coding for Elementary Music Learning	Park & Kwon [11]
2	research, methodological, textual, data, framework, proposes, classical, ancient, historical, archive	Vibe Coding for Classical Text Research and Archiving	Kim [52]
3	educational, education, tools, development, career, language, students, analysis, tool, design	Vibe Coding for Career and Language Education	Lee & Hwang [8]
4	design, learning, students, program, instructional, app, elementary, problemsolving, principles, generative	Vibe Coding for Elementary Instructional Design	Lee et al. [53]
5	literacy, digital, learning, analysis, research, literary, data, tools, application, design	Vibe Coding for Digital and Data-Driven Research	Oh [54]
6	chinese, education, educational, design, classical, research, development, tools, tool, digital	Vibe Coding for Classical Chinese Education	Jung [55]
7	development, scaffolding, students, writing, chatbots, english, design, code, data, software	Vibe Coding for AI-Scaffolded Learning	Shin & Yoon [56]

Fig. 7. Intertopic Distance Map of Korean Literature Related to Vibe Coding

토픽 1은 '초등 음악 학습을 위한 바이브 코딩(Vibe coding for elementary music learning)'으로 명명했다. learning, music, elementary, web 등의 단어가 주요하게 나타났으며, 대표 논문[11]에서는 초등교사가 바이브 코딩을 사용해 음악 수업을 위한 5개의 게임 기반 웹 애플리케이션을 개발한 사례를 제시했다.

토픽 2는 '고전 텍스트 연구와 아카이빙을 위한 바이브 코딩(Vibe coding for classical text research and archiving)'으로 명명했다. methodological, textual, classical, ancient, historical, archive 등의 단어가 도출되었고, 대표 논문[52]은 고전 텍스트와 관련 자료를 대상으로 연구와 분석에 활용할 수 있는 디지털 아카이브 구축의 방법론적 가능성을 다루었다.

토픽 3은 '진로 및 언어교육을 위한 바이브 코딩(Vibe coding for career and language education)'으로 명명했다. career, language, educational, students, development 등의 주요 단어가 나타났으며, 대표 논문[8]은 대학 정규 교과과정에서 LLM 기반 플랫폼을 활용해 진로상담 챗봇을 설계하는 실습을 운영하고 참여적 실행연구를 통해 학습자 경험을 분석했다.

토픽 4는 '초등 교수설계를 위한 바이브 코딩(Vibe coding for elementary instructional design)'으로 명명했다. 주요 단어는 design, instructional, elementary, learning, students 등이었고, 대표 논문[53]은 초등교육 맥락에서 생성형 AI와 바이브 코딩을 활용한 교수·학습 프로그램의 설계와 적용을 다루었다.

토픽 5는 '디지털 및 데이터 기반 연구를 위한 바이브 코딩(Vibe coding for digital and data-driven research)'으로 명명했다. digital, analysis, research, data, tools, application 등의 단어가 주요하게 나타났으며, 대표 논문[54]에서는 문화유산 조사와 관련된 AI 기반 데이터 분석 및 디지털 전환의 가능성을 검토하고 바이브 코딩을 활용한 현장 지원 애플리케이션 개발 가능성을 제안했다.

토픽 6은 '한문교육을 위한 바이브 코딩(Vibe coding for classical Chinese education)'으로 명명했다. chinese, classical, education 등의 단어가 두드러지며, 대표 논문[55]은 한문교육 및 예비 한문 교사 교육 맥락에서 바이브 코딩과 생성형 AI가 교과 특성에 맞는 교수·학습 자료 및 디지털 교육 도구를 개발하는 데 활용될 가능성을 보여주었다.

토픽 7은 'AI 기반 학습 스캐폴딩을 위한 바이브 코딩(Vibe coding for AI-scaffolded learning)'으로 명명했다. scaffolding, students, development, design, software 등의 단어가 나타났으며, 대표 논문[56]에서는

www.kci.go.kr

바이브 코딩을 활용해 티볼 전략 학습 지원 애플리케이션을 개발하고 소크라테스식 질문을 통한 인지적 스캐폴딩을 구현했다.

종합하면, 국제 연구에서는 의학·임상 분야를 중심으로 바이브 코딩을 활용한 소프트웨어 및 교육 도구 개발과 함께 개발 과정의 책임성 및 거버넌스가 주요 연구 주제로 나타났으나, 국내 연구는 특정 교과와 교수학습 맥락에 바이브 코딩을 적용해 디지털 도구와 학습 지원 환경을 개발하는 연구가 상대적으로 많았다. 이는 국내외 바이브 코딩 연구가 공통으로 LLM을 활용한 소프트웨어 개발의 가능성을 탐색하면서도, 국제 연구는 거시적 관점에서 개발 및 활용 방식과 관리 문제에, 국내 연구는 미시적 관점에서 현장의 구체적인 적용에 더 초점을 두었음을 보여준다.

IV. Conclusions

본 연구는 바이브 코딩과 관련된 국내외 학술 연구의 전반적인 양상과 주요 연구 주제를 비교·분석하기 위해 2025년 이후 발표된 국내외 논문 83편을 대상으로 TF-IDF 기반 키워드의 시계열 분석과 LDA 기반 토픽 모델링을 수행했다. 분석 결과 바이브 코딩에 관한 논문은 2025년 이후 증가하는 양상을 보였으며, 국제 연구와 국내 연구 모두 바이브 코딩의 다양한 측면을 탐색하면서도 서로 다른 연구 주제와 적용 맥락을 다루었다. 국제 연구는 특정 분야 적용 사례와 개발 가능성 탐색에서 시작해 바이브 코딩을 하나의 소프트웨어 개발 접근법으로 논의하고, 실제 개발 과정과 코드 생성 및 자동화, 책임성 및 거버넌스 등의 문제로 연구 관심이 확장되었다. 국내 연구는 교육 및 학습 맥락을 중심으로 바이브 코딩을 활용한 디지털 도구 개발이 이루어졌으며, 학생과 학습자의 경험, 교사와 교육 실천가의 직접적인 설계 및 개발 활동 등 구체적 사례 중심의 연구가 이루어졌다.

이러한 결과는 본 연구의 이론적 배경에서 살펴본 바와 같이, 인간과 시스템의 역할이 변화하는 프로그래밍 추상화의 연장선으로서 바이브 코딩을 다루는 관점에서 논의할 수 있다. 국제 연구에서 개발 자동화와 책임성·거버넌스 등이 주요 주제로 나타난 점은 구현 과정의 일부가 시스템에 위임되는 것과 동시에 생성 결과를 평가하고 관리하는 것이 인간의 역할이 될 가능성을 보여준다. 국내 연구에서 학생, 학습자 및 교사가 바이브 코딩으로 교육용 도구를 직접 설계·개발하는 사례가 지속적으로 나타난 것은 프로그래밍 추상화 과정에서의 개발 참여 주체 확대가 바이브 코딩 연구에서도 이어지고 있는 것으로 해석된다. 이 관점에서 향후 바이브 코딩 연구는 단순히 시스템이 어느 수준까지 개발 작업을 수행할 수 있는지 확인하는 데 그치지 않고, 개발 과정에서 인간과 시스템이 각각 어떠한 역할을 수행하며 그 역할이 어떻게 변화하는지를 함께 살펴볼 필요가 있다. 특히 개발 참여 주체가 확대될수록 비프로그래머가 자신의 목적에 맞는 결과물을 생성할 수 있도록 지원하는 방안과 함께, 생성된 결과의 적절성 및 신뢰성을 평가하고 검증하는 방안에 관한 논의도 필요하다.

본 연구는 바이브 코딩에 관한 초기 학술 연구의 전반적 연구 지형을 국내외 비교를 통해 파악했다는 의의가 있다. 그러나 바이브 코딩이라는 용어가 등장한 이후의 연구를 대상으로 했기 때문에 분석 기간이 짧고 문헌 수가 제한적이며, 문헌 수집 과정에서 바이브 코딩과 유사한 현상을 다른 용어로 다룬 연구가 포함되지 않았을 가능성이 있다. 또한 초록을 중심으로 텍스트 분석을 수행했기 때문에 개별 연구의 구체적인 방법과 맥락을 충분히 반영하지 못했다. 더불어 본 연구의 단독 저자만이 분석에 참여해 문헌 선정 및 분석 결과에 대한 타당성을 충분히 검증하지는 못했다는 한계를 가진다. 후속 연구에서는 분석 대상과 기간을 확대하고, 실제 바이브 코딩의 개발 과정과 사용자 경험을 질적으로 검토할 필요가 있다. 본 연구가 바이브 코딩을 비롯한 새로운 프로그래밍 방식의 변화와 그에 따른 인간과 시스템의 관계를 이해하는 기초 자료가 되고, 더 많은 사람이 자신의 지식과 아이디어를 디지털 기술로 구현하며 사회의 다양한 문제 해결에 참여할 수 있는 미래를 탐색하는 토대가 되길 소망한다.

REFERENCES

[1] X. Hou, J. Zhao, Y. Liu, Z. Yang, Y. Wang, H. Li, and H. Zhang, “Large language models for software engineering: A systematic literature review,” ACM Transactions on Software Engineering and Methodology, Vol. 33, No. 8, Article 220, Dec. 2024. DOI: 10.1145/3695988

[2] C. Meske, T. Hermanns, E. Von der Weiden, K. U. Loser, and T. Berger, “Vibe coding as a reconfiguration of intent mediation in software development: Definition, implications, and research agenda,” IEEE Access, Vol. 13, pp. 213242-213259, Dec. 2025. DOI: 10.1109/ACCESS.2025.3645466

[3] S. Siddeeq, M. Waseem, K. K. Kemell, M. Saari, J. Rasku, and P. Abrahamsson, “Vibe coding in software development: A multivocal literature review,” arXiv, July 2026. DOI: 10.48550/arXiv.2607.21652

www.kci.go.kr

[4] F. Pesce, and W. Cheungpasitporn, “Vibe coding in nephrology education: Clinician-led, AI-assisted development of open-source interactive learning tools,” Renal Failure, Vol. 47, No. 1, Article 2581933, Nov. 2025. DOI: 10.1080/0886022X.2025.2581933

[5] Y. S. Kıyak, E. Emekli, T. İ. Kara, Ö. Coşkun, and I. İ. Budakoğlu, “AI teaches surgical diagnostic reasoning to medical students: Evidence from an experiment using a fully automated, low-cost feedback system,” Journal of Surgical Education, Vol. 82, No. 10, Article 103639, Oct. 2025. DOI: 10.1016/j.jsurg.2025.103639

[6] J. W. Park, “A study on developing a vibe coding-based Chinese language learning game web app,” Journal of Sinology and China Studies, Vol. 104, pp. 3-30, Sept. 2025. DOI: 10.18077/chss.2025.104..001

[7] R. Rahman, “Vibe coding in nursing education: From user to creator,” Nurse Education Today, Vol. 166, Article 107278, Nov. 2026. DOI: 10.1016/j.nedt.2026.107278

[8] H. E. Lee, and H. A. Hwang, “Action research on an AI career counseling chatbot design practicum: Focusing on large language models (LLMs),” Korea Journal of Counseling, Vol. 27, No. 1, pp. 283-310, Feb. 2026. DOI: 10.15703/kjc.27.1.202602.283

[9] J. G. Meyer, “Vibe coding omics data analysis applications,” Journal of Proteome Research, Vol. 25, No. 2, pp. 1191-1197, Jan. 2026. DOI: 10.1021/acs.jproteome.5c00984

[10] H. J. Lee, and D. J. Kim, “Educational effects of generative AI and no-code vibe coding classes: An analysis of non-CS students’ project outputs,” Journal of Next-generation Convergence Technology Association, Vol. 9, No. 11, pp. 3220-3228, Nov. 2025. DOI: 10.33097/JNCTA.2025.09.11.3220

[11] S. I. Park, and H. K. Kwon, “Design and development of an AI-based vibe coding web app for elementary music learning,” Korean Journal of Research in Music Education, Vol. 55, No. 2, pp. 167-191, May 2026. DOI: 10.30775/KMES.55.2.167

[12] A. Fawzy, A. Tahir, and K. Blincoe, “Vibe coding in practice: Motivations, challenges, and a future outlook–a grey literature review,” Proceedings of the IEEE/ACM 48th International Conference on Software Engineering: Software Engineering in Practice, pp. 212-223, Rio de Janeiro, Brazil, July 2026. DOI: 10.1145/3786583.3786866

[13] L. A. Miller, “Programming by non-programmers,” International Journal of Man-Machine Studies, Vol. 6, No. 2, pp. 237-260, March 1974. DOI: 10.1016/S0020-7373(74)80004-0

[14] L. A. Miller, “Natural language programming: Styles, strategies, and contrasts,” IBM Systems Journal, Vol. 20, No. 2, pp. 184-215, Dec. 1981. DOI: 10.1147/sj.202.0184

[15] L. A. Onorato, and R. W. Schvaneveldt, “Programmer-nonprogrammer differences in specifying procedures to people and computers,” Journal of Systems and Software, Vol. 7, No. 4, pp. 357-369, Dec. 1987. DOI: 10.1016/0164-1212(87)90034-3

[16] A. Ebrahimi, “Novice programmer errors: Language constructs and plan composition,” International Journal of Human-Computer Studies, Vol. 41, No. 4, pp. 457-480, Oct. 1994. DOI: 10.1006/ijhc.1994.1069

[17] B. A. Myers, J. F. Pane, and A. J. Ko, “Natural programming languages and environments,” Communications of the ACM, Vol. 47, No. 9, pp. 47-52, Sept. 2004. DOI: 10.1145/1015864.1015888

[18] S. Rivard, and S. L. Huff, “Factors of success for end-user computing,” Communications of the ACM, Vol. 31, No. 5, pp. 552-561, May 1988. DOI: 10.1145/42411.42418

[19] J. C. Brancheau, and C. V. Brown, “The management of end-user computing: Status and directions,” ACM Computing Surveys, Vol. 25, No. 4, pp. 437-482, Dec. 1993. DOI: 10.1145/162124.162138

[20] G. Fischer, E. Giaccardi, Y. Ye, A. G. Sutcliffe, and N. Mehandjiev, “Meta-design: A manifesto for end-user development,” Communications of the ACM, Vol. 47, No. 9, pp. 33-37, Sept. 2004. DOI: 10.1145/1015864.1015884

[21] A. I. Mørch, G. Stevens, M. Won, M. Klann, Y. Dittrich, and V. Wulf, “Component-based technologies for end-user development,” Communications of the ACM, Vol. 47, No. 9, pp. 59-62, Sept. 2004. DOI: 10.1145/1015864.1015890

[22] F. Modugno, A. T. Corbett, and B. A. Myers, “Graphical representation of programs in a demonstrational visual shell—an empirical evaluation,” ACM Transactions on Computer-Human Interaction, Vol. 4, No. 3, pp. 276-308, Sept. 1997. DOI: 10.1145/264645.264659

[23] J. Chen, and A. Zelinsky, “Programming by demonstration: Coping with suboptimal teaching actions,” The International Journal of Robotics Research, Vol. 22, No. 5, pp. 299-319, May 2003. DOI: 10.1177/0278364903022005002

[24] D. Wolber, “Pavlov: An interface builder for designing animated interfaces,” ACM Transactions on Computer-Human Interaction, Vol. 4, No. 4, pp. 347-386, Dec. 1997. DOI: 10.1145/267135.267142

[25] J. R. Chen, “Constructing task-level assembly strategies in robot programming by demonstration,” The International Journal of Robotics Research, Vol. 24, No. 12, pp. 1073-1085, Dec. 2005. DOI: 10.1177/0278364905060120

[26] V. Castelli, D. Oblinger, and L. Bergman, “Augmentation-based learning combining observations and user edits for programming-by-demonstration,” Knowledge-Based Systems, Vol. 20, No. 6, pp. 575-591, Aug. 2007. DOI: 10.1016/j.knosys.2007.04.007

[27] H. Khalajzadeh, and J. Grundy, “Accessibility of low-code approaches: A systematic literature review,” Information and Software Technology, Vol. 177, Article 107570, Jan. 2025. DOI: 10.1016/j.infsof.2024.107570

[28] S. Matook, Y. M. Wang, and M. Axelsen, “Experiential learning for citizen developers,” Business & Information Systems

www.kci.go.kr

Engineering, Vol. 67, pp. 7-30, Feb. 2025. DOI: 10.1007/s12599-024-00921-3

[29] A. Viljoen, B. Stelzl, M. Yang, J. Nguyen, A. Hein, E. Elshan, and H. Krcmar, "Navigating flexibility and standardisation in low-code/no-code development," Information Systems Journal, Vol. 36, No. 1, pp. 95-109, June 2026. DOI: 10.1111/isj.70001

[30] S. Rafi, M. A. Akbar, and S. Mahmood, "Low-code and no-code tools: A valuable aid or a potential threat to professional developers?" Software: Practice and Experience, Vol. 56, No. 6, pp. 802-819, Apr. 2026. DOI: 10.1002/spe.70067

[31] Y. Liu, C. Tantithamthavorn, Y. Liu, and L. Li, "On the reliability and explainability of language models for program generation," ACM Transactions on Software Engineering and Methodology, Vol. 33, No. 5, pp. 1-26, June 2024. DOI: 10.1145/3641540

[32] Y. Lee, and S. Huh, "How can clinicians leverage vibe coding for machine learning and deep learning research?" Endocrinology and Metabolism, Vol. 40, No. 5, pp. 659-667, Oct. 2025. DOI: 10.3803/EnM.2025.2675

[33] J. Oertel, J. Klünder, and R. Hebig, "Don't settle for the first! How many GitHub Copilot solutions should you check?" Information and Software Technology, Vol. 183, Article 107737, July 2025. DOI: 10.1016/j.infsof.2025.107737

[34] F. Tambon, A. Moradi-Dakhel, A. Nikanjam, F. Khomh, M. C. Desmarais, and G. Antoniol, "Bugs in large language models generated code: An empirical study," Empirical Software Engineering, Vol. 30, No. 3, Article 65, Feb. 2025. DOI: 10.1007/s10664-025-10614-4

[35] M. Chow, and O. Ng, "From technology adopters to creators: Leveraging AI-assisted vibe coding to transform clinical teaching and learning," Medical Teacher, Vol. 47, No. 12, pp. 1927-1929, Apr. 2025. DOI: 10.1080/0142159X.2025.2488353

[36] O. Ng, and M. Chow, "Empowering health professions educators: Developing educational tools with AI-assisted vibe coding," Medical Science Educator, Vol. 36, pp. 47-52, Feb. 2026. DOI: 10.1007/s40670-025-02596-1

[37] B. R. Barricelli, F. Cassano, D. Fogli, and A. Piccinno, "End-user development, end-user programming and end-user software engineering: A systematic mapping study," Journal of Systems and Software, Vol. 149, pp. 101-137, March 2019. DOI: 10.1016/j.jss.2018.11.041

[38] M. Santos, and M. L. B. Villela, "Characterizing end-user development solutions: A systematic literature review," Human-Computer Interaction. Perspectives on Design, pp. 194-209, Orlando, USA, June 2019. DOI: 10.1007/978-3-030-22646-6_14

[39] A. J. Ko, R. Abraham, L. Beckwith, A. Blackwell, M. Burnett, M. Erwig, C. Scaffidi, J. Lawrance, H. Lieberman, B. Myers, M. B. Rosson, G. Rothermel, M. Shaw, and S. Wiedenbeck, "The state of the art in end-user software engineering," ACM Computing Surveys, Vol. 43, No. 3, Article 21, Apr. 2011. DOI: 10.1145/1922649.1922658

[40] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," Robotics and Autonomous Systems, Vol. 57, No. 5, pp. 469-483, May 2009. DOI: 10.1016/j.robot.2008.10.024

[41] G. Ajaykumar, M. Steele, and C.-M. Huang, "A survey on end-user robot programming," ACM Computing Surveys, Vol. 54, No. 8, Article 164, Oct. 2021. DOI: 10.1145/3466819

[42] A. Bucaioni, A. Cicchetti, and F. Ciccozzi, "Modelling in low-code development: A multi-vocal systematic review," Software and Systems Modeling, Vol. 21, No. 5, pp. 1959-1981, Jan. 2022. DOI: 10.1007/s10270-021-00964-0

[43] B. Binzer, and T. J. Winkler, "Democratizing software development: A systematic multivocal literature review and research agenda on citizen development," Software Business, pp. 244-259, Bolzano, Italy, Oct. 2022. DOI: 10.1007/978-3-031-20706-8_17

[44] M. O. Ajimati, N. Carroll, and M. Maher, "Adoption of low-code and no-code development: A systematic literature review and future research agenda," Journal of Systems and Software, Vol. 222, Article 112300, Apr. 2025. DOI: 10.1016/j.jss.2024.112300

[45] T. H. Min, and B. W. Lee, "Research trends in measuring digital literacy of science teachers: Focusing on bibliometric analysis and theoretical framework analysis," New Physics: Sae Mulli, Vol. 75, No. 1, pp. 74-86, Jan. 2025. DOI: 10.3938/NPSM.75.74

[46] T. H. Min, Y. J. Moon, and H. J. Maeng, "Comparison of domestic and international AI literacy research topics using text mining techniques," Journal of the Korea Society of Computer and Information, Vol. 30, No. 2, pp. 213-226, Feb. 2025. DOI: 10.9708/JKSCI.2025.30.02.213

[47] I. A. Elgendy, Y. K. Dwivedi, M. A. Al-Sharafi, M. Hosny, M. Y. I. Helal, T. Crick, L. Hughes, S. Alwahaishi, M. Mahmud, V. Dutot, and A. S. Al-Busaidi, "Responsible vibe coding: Architecture, opportunities, and research agenda," Journal of Computer Information Systems, pp. 1-19, Jan. 2026. DOI: 10.1080/08874417.2026.2621186

[48] A. Abendroth, "The democratization of software engineering: Evolution, definition, and the future of low-code and no-code platforms," Management Review Quarterly, pp. 1-36, June 2026. DOI: 10.1007/s11301-026-00603-2

[49] H. V. Tran, D. A. Nguyen, and P. N. Hung, "VC-AWG: A vibe coding–based method for automated website generation from heterogeneous artifacts," Journal of Computer Languages, Vol. 88, Article 101403, Sept. 2026. DOI: 10.1016/j.cola.2026.101403

[50] Y. Otsuki, and T. Nuri, "Breast reconstruction morphology evaluation tool: Vibe coding empowers clinicians as creators," Plastic and Reconstructive Surgery Global Open, Vol. 14, No. 5, Article e7691, May 2026. DOI: 10.1097/GOX.000000000000

www.kci.go.kr

7691

[51] J. E. García, “Vibe-coding as a teaching competency in the use of artificial intelligence for medical education: A scoping review,” Revista Española de Educación Médica, Vol. 7, No. 2, Article 2, March 2026. DOI: 10.6018/edumed.705121

[52] Y. J. Kim, “Designing a light archive as a research methodology for classical text utilization—focusing on life-thought texts held in the Jongyeonggak Collection: A methodological experiment grounded in the Jongyeonggak Collection,” Daedong Munhwa Yeongu, Vol. 133, pp. 179-208, March 2026. DOI: 10.18219/ddmh..133.202603.179

[53] C. S. Lee, Y. K. Yu, and S. W. Sung, “Exploring elementary practical arts class models integrating generative AI and educational robots: From block-based coding to vibe coding,” Journal of Korean Practical Arts Education Research, Vol. 32, No. 1, pp. 47-84, Feb. 2026. DOI: 10.29113/skpaer.2026.32.1.03

[54] D. Y. Oh, “Methodologies of data-driven hybrid archaeology: Theory and practice of AI-integrated buried heritage investigations,” Humanities Contents, Vol. 80, pp. 167-190, March 2026. DOI: 10.18658/humancon.2026.03.167

[55] H. J. Jung, “Direction of classical Chinese education for training teachers in the AI era,” Journal of Chinese Characters Education in Korea, Vol. 66, pp. 29-60, June 2026. DOI: 10.17963/ccek.2026.66.29

[56] J. G. Shin, and S. Y. Yoon, “Development of a T-ball strategy learning support application utilizing Socratic generative AI,” The Korean Journal of Physical Education, Vol. 65, No. 3, pp. 717-732, May 2026. DOI: 10.23949/kjpe.2026.5.65.3.49

Authors

Tae-Ho Min received the M.S. degree in Science Education from Seoul National University, Korea, in 2023, and the Ph.D. degree in Science Education from Dankook University, Korea, in 2026.

He is currently a physics teacher at Bora High School, Korea. His current research interests include artificial intelligence in education (AIED) and educational data mining (EDM).

www.kci.go.kr