

사용자의 AI 코딩 에이전트 활용 능력 측정을 위한 지표 개발 및 분석 도구 제안

(Proposal for Metrics and Analysis Tools to Measure User
Proficiency with AI Coding Agents)

지도교수 : 유승주

이 보고서를 공학학사 학위 논문
대체 보고서로 제출함.

2025년 11월 27일

서울대학교 공과대학
컴퓨터공학부
김상민

2026년 2월

국문초록

주요어: AI 코딩 에이전트, AI 활용 능력, 지표 개발, RetroChat, LLM-as-a-judge, 루브릭

AI 코딩 에이전트가 소프트웨어 개발의 새로운 패러다임으로 자리 잡음에 따라, 사용자가 얼마나 효과적으로 AI와 협업하는지 측정할 필요성이 대두되고 있다. 기존의 분석 도구들은 비용이나 토큰 사용량과 같은 단순한 정량적 지표에 머무르는 한계가 있었다. 본 연구는 이러한 한계를 극복하기 위해 두 가지 핵심 요소를 제안한다.

첫째, 다양한 상용 AI 코딩 에이전트의 채팅 로그를 다면적으로 수집하고 분석하는 오픈소스 툴킷 'RetroChat'을 개발한다.

둘째, 수집된 채팅 히스토리 원천 데이터를 기반으로, 'LLM-as-a-judge' 방법론을 활용하여 사용자의 AI 상호작용 품질을 평가하는 시스템을 구축한다. 본 연구에서는 토큰 효율성과 같은 객관적 지표를 기준으로 우수 세션을 선별하고, LLM을 통해 해당 세션들의 성공 요인을 학습하여 평가 루브릭(rubric)을 자동으로 도출하는 지도학습 기반의 파이프라인을 제안한다. 이 시스템은 사용자의 'AI 활용 능력(AI Proficiency)'을 루브릭에 따라 다각도로 점수화함으로써, 개발자 교육 및 AI 협업 프로세스 개선에 기여하는 것을 목표로 한다.

Contents

국문초록	I
1 서론	1
2 관련 연구	2
2.1 AI 협업 데이터 분석 툴링	2
2.2 채팅 히스토리 정성 평가 방법론	3
3 접근 방식	4
3.1 RetroChat: 다면적 AI 코딩 채팅 분석 툴킷	4
3.1.1 구현	4
3.1.2 주요 기능 및 특징	4
3.2 LLM-as-a-judge 기반 사용자 AI 활용 능력 평가	6
3.2.1 평가 시스템 구축 과정	7
3.2.2 지도 학습 기반의 AI 활용 능력 루브릭(Rubric) 자동 도출: 훈련 파이프라인 구현 세부	8
4 평가	10
4.1 평가 및 학습용 데이터셋 구축	10
4.1.1 원천 데이터 구성	10
4.1.2 세션별 정답 스코어 정의	10
4.2 지도학습 결과	10

4.2.1 토큰 효율 기준 루브릭	11
4.2.2 사용자 턴 효율 기준 루브릭	12
4.3 검증 결과	13
4.3.1 토큰 효율 기준 검증 결과	15
4.3.2 사용자 턴 효율 기준 검증 결과	15
4.3.3 척도 간 상관관계 해석	16
4.3.4 낮은 상관관계의 원인 분석 및 개선 방향	16
5 결론	18
부록	21
5.1 Rubric Extraction Prompt Template	21
5.2 Rubric Summarization Prompt Template	22
5.3 Token Efficiency Rubrics	22
5.3.1 LLM Summarization Method (3 rubrics)	22
5.3.2 HDBSCAN Clustering Method (5 rubrics)	24
5.4 User Turn Efficiency Rubrics	26
5.4.1 LLM Summarization Method (3 rubrics)	26
5.4.2 HDBSCAN Clustering Method (5 rubrics)	28

1. 서론

최근 소프트웨어 개발 분야에서 'AI Vibe Coding'으로 대표되는 AI 코딩 에이전트와의 협업은 강력한 선택지가 되어가고 있다. 이러한 변화 속에서 개발자가 AI 도구를 얼마나 '잘' 활용하는지가 생산성에 지대한 영향을 미치기 시작했다. 하지만 사용자의 AI 활용 능력을 객관적으로 측정하고 평가하는 표준화된 지표나 도구는 매우 부족한 실정이다.

시장에 공개된 일부 AI 협업 데이터 분석 도구들은 존재하지만, 대부분 비용, 토큰 사용량 같은 단순 정량 지표에 국한된다. 이러한 지표들은 사용자가 얼마나 많이 AI를 사용했는지는 보여줄 수 있으나, 얼마나 효과적으로 문제를 해결하고, 얼마나 질 높은 상호작용을 했는지는 파악하기 어렵다.

본 연구는 이러한 문제의식에서 출발하여, 사용자의 AI 코딩 에이전트 활용 능력을 심층적으로 측정하기 위한 새로운 접근 방식을 제안한다. 이를 위해 본 연구는 두 가지 주요한 기여를 한다.

첫째, Claude Code, Gemini CLI, Codex 등 다양한 벤더사의 AI 코딩 에이전트로부터 채팅 내역, 비용, 토큰 사용량, 파일 변경 내역 등 다면적인 데이터를 수집하고 아래 점수화 시스템을 통해 리포트화하는 오픈소스 툴킷 'RetroChat'을 개발한다.

둘째, 'RetroChat'을 통해 수집된 채팅 히스토리를 원천 데이터로 삼아, 'LLM-as-a-judge' 방법론을 적용한다. 이는 강력한 LLM(대형 언어 모델)을 '평가자'로 활용하는 과정으로, 단순히 사전에 정의된 기준으로 평가하는 것을 넘어, 지도학습(Supervised learning) 기반의 자동화된 루브릭(rubric) 도출 파이프라인을 구축하는 것을 핵심으로 한다. 이 파이프라인은 토큰 효율성 등 정량적 지표가 높은 채팅 세션들을 학습 데이터로 사용하여, 성공적인 상호작용의 핵심 패턴을 LLM 스스로 학습하고 이를 평가 루브릭으로 생성한다. 이후, 생성된 루브릭을 기반으로 새로운 상호작용의 질을 다각도로 점수화하는 시스템을 구축한다.

본 보고서는 먼저 관련 연구들을 살펴본 뒤, 우리가 제안하는 'RetroChat' 툴킷과 'LLM-as-a-judge' 기반의 AI 활용 능력 평가 시스템의 설계 및 구현 방식을 상세히 설명한다.

2. 관련 연구

본 연구는 'AI 협업 데이터 분석'과 '채팅 히스토리 정성 평가'라는 두 가지 주요 영역의 기존 연구들을 기반으로 한다.

2.1 AI 협업 데이터 분석 툴링

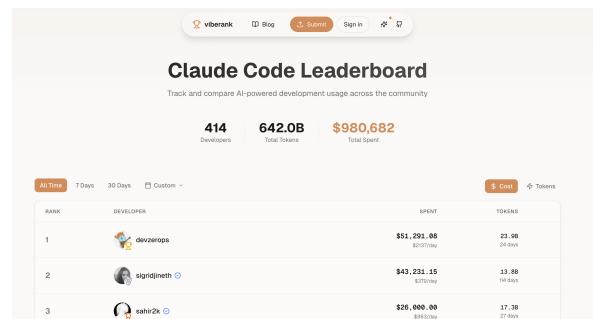
AI와의 협업 개발 형태가 보편화되면서, 이러한 상호작용 데이터를 회고하고 분석하려는 시도 들이 등장하고 있다. 대표적으로 ryoppippi/ccusage[1]나 sculptdotfun/viberank[2]와 같은 오픈 소스 프로젝트들이 있다.

ryoppippi/ccusage: 주로 AI 사용에 따른 비용(Cost)과 토큰(Token) 사용량을 추적하고 시각화하는 데 중점을 둔다. 이는 조직이나 개인이 AI 활용에 드는 비용을 관리하는 데 유용하다.

sculptdotfun/viberank: AI 사용에 따른 토큰(Token) 사용량을 기반으로, 사용량을 달리 단위의 비용으로 변환하여 보여주는 대시보드 사이트이다.



(a) ccusage



(b) viberank

Figure 2.1: ccusage 사용자 화면 및 viberank 사용자 화면

이러한 도구들은 AI 협업의 특정 단면(주로 비용 및 사용량)을 정량적으로 파악하는 데 유용성을 가지나, 사용자가 AI에 얼마나 명확하게 요구사항을 전달했는지, AI의 제안을 얼마나 비판적으로 수용하고 개선했는지와 같은 상호작용의 '질'을 평가하는 데는 명확한 한계가 있다.

2.2 채팅 히스토리 정성 평가 방법론

AI와의 채팅 히스토리를 정성적으로 평가하려는 연구도 활발히 진행되고 있다. 그중 본 연구와 밀접하게 관련된 것은 'SPUR'[3] 방법론이다.

SPUR (Supervised Prompting for User satisfaction Rubrics): 이 연구는 사용자의 '호/불호'가 레이블링된 정성적인 채팅 히스토리 데이터셋을 활용한다. 지도 학습(Supervised Learning) 형태를 통해, 사용자가 특정 채팅 내역을 선호하거나 선호하지 않는 이유가 되는 다양한 측면의 피쳐(루브릭)를 모델이 스스로 학습하여 뽑아내도록 한다. 이렇게 학습된 모델은 새로운 채팅 히스토리가 입력되었을 때, 해당 상호작용에 대한 사용자의 잠재적인 호/불호를 예측하고 그 근거를 제시한다.

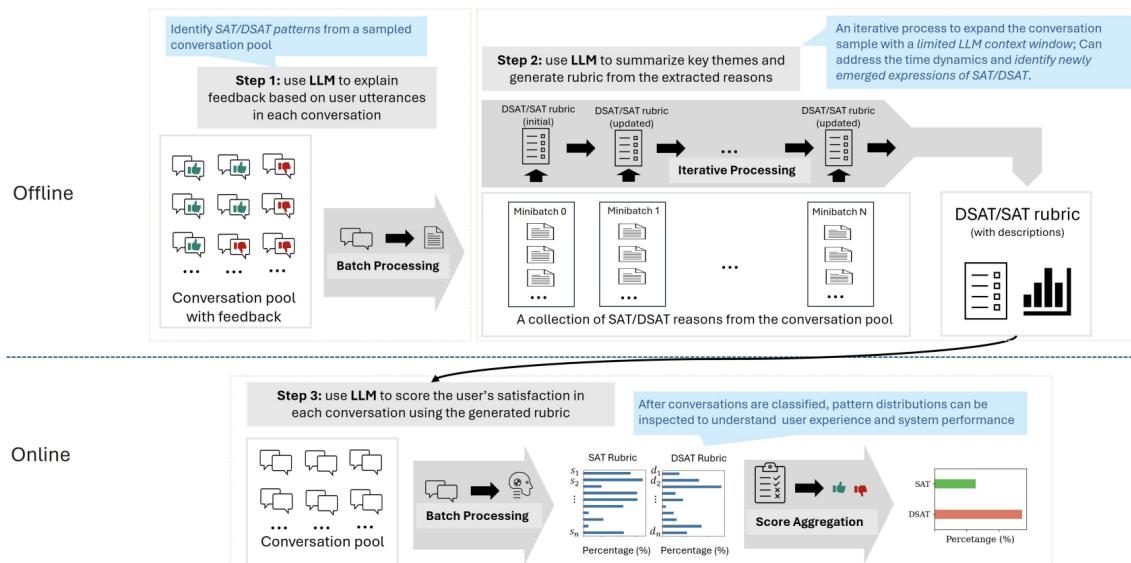


Figure 2.2: SPUR 방법론의 지도학습 과정 / 평가 과정

SPUR는 LLM을 활용해 채팅의 정성적 품질을 평가하는 루브릭을 도출할 수 있다는 가능성을 보여주었다. 하지만 이는 사용자의 '선호도' 예측에 초점을 맞추고 있다. 본 연구는 여기서 더 나아가, 선호도를 넘어 다양한 정량 지표를 바탕으로 사용자의 '활용 능력'을 판단하는 루브릭을 개발하고 이를 점수화하는 시스템을 구축하고자 한다.

3. 접근 방식

본 연구의 접근 방식은 크게 두 단계로 구성된다. 첫째, AI 코딩 에이전트와의 상호작용 데이터를 다각도로 수집하기 위한 툴킷 'RetroChat'을 개발한다. 둘째, 수집된 데이터를 기반으로 'LLM-as-a-judge' 방법론을 활용하여 사용자의 AI 활용 능력을 평가하는 시스템을 구축한다.

3.1 RetroChat: 다면적 AI 코딩 채팅 분석 툴킷

사용자의 AI 활용 능력을 분석하기 위해서는 먼저 신뢰할 수 있는 원천 데이터의 확보가 필수적이다. 기존 도구들이 제공하는 정량적 지표만으로는 심층적인 분석이 불가능하다고 판단하여, 다면적인 정보를 수집할 수 있는 오픈소스 툴킷 'RetroChat'¹을 설계 및 구현했다.

3.1.1 구현

'RetroChat'은 Rust 프로그래밍 언어와 Tauri 프레임워크를 기반으로 제작된 Desktop GUI 애플리케이션이다. 이는 직관적인 그래픽 인터페이스를 제공하여 사용자가 자신의 AI 상호작용 데이터를 쉽게 탐색하고 분석할 수 있도록 돕는다. 수집된 모든 사용자 데이터셋은 로컬 SQLite 데이터베이스에 저장되어, 데이터의 프라이버시를 보장하면서 효율적으로 관리된다.

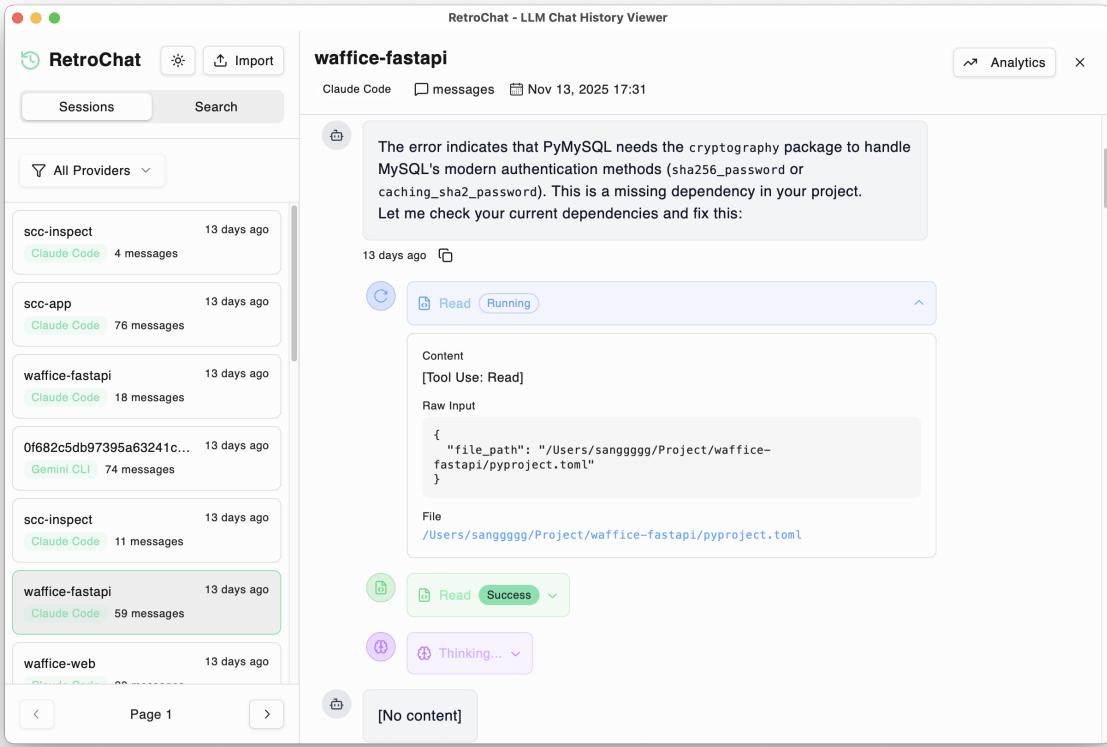
3.1.2 주요 기능 및 특징

1. **다면적 데이터 수집:** 'RetroChat'은 단순한 채팅 로그(Chat History) 스크래핑을 넘어, AI 상호작용과 관련된 다양한 맥락 정보를 수집한다.

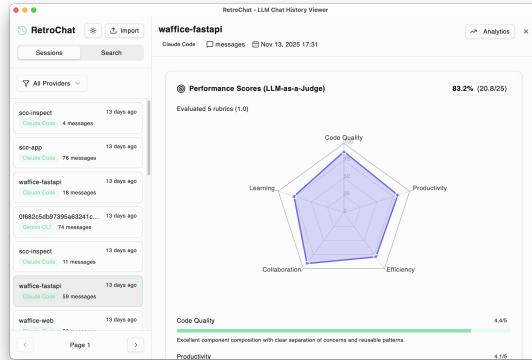
- **Cost & Token Usage:** API 호출에 따른 비용 및 토큰 사용량.
- **File Changes:** AI의 제안이 실제 프로젝트 파일에 어떤 변경(추가, 삭제, 수정)을 가져왔는지 추적.
- **Chat History:** 사용자와 AI 간의 전체 대화 내용.
- **Tool Use:** AI 가 사용한 Tool 종류 별 사용 횟수.

¹<https://github.com/wafflestudio/retrochat>

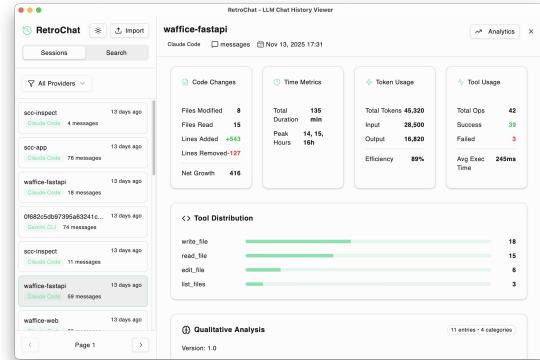
2. **다양한 벤더사 지원:** 협업에서 사용되는 다양한 AI 코딩 에이전트를 지원하는 것을 목표로 한다. 현재 Claude Code, Gemini CLI, Codex 등 주요 벤더사의 채팅 로그 형식을 파싱하고 분석할 수 있다.
3. **LLM-as-a-judge 기반 리포트화:** 수집된 원천 데이터를 단순히 나열하는 것이 아니라, 후술 할 3.2절의 LLM-as-a-judge 방법론을 내장하여, 수집된 데이터를 바탕으로 사용자의 상호작용 패턴을 분석하고 평가하는 요약 리포트를 생성한다.



(a) 채팅 세션 목록 화면



(b) 분석 결과 화면 1



(c) 분석 결과 화면 2

Figure 3.1: RetroChat 사용자 화면. 채팅 세션 목록 및 분석 결과 화면

3.2 LLM-as-a-judge 기반 사용자 AI 활용 능력 평가

'RetroChat'을 통해 수집된 채팅 히스토리 데이터는 사용자의 AI 활용 능력을 평가하기 위한 핵심 원천 데이터가 된다. 본 연구는 이 정성적인 텍스트 데이터를 평가하기 위해 'LLM-as-a-

judge’ 방법론²을 도입했다. 이는 관련 연구인 SPUR가 사용자의 ’선호도’를 예측한 것과 달리, 사용자의 ’활용 능력(Proficiency)’을 학습 시점에 정량적으로 직접 정의하고 측정 및, 긍정적이거나 부정적인 부분을 판단하여 사용자에게 전달하는 것에 중점을 둔다.

3.2.1 평가 시스템 구축 과정

1. **지도 학습 기반의 AI 활용 능력 루브릭(Rubric) 자동 도출 파이프라인:** 본 연구는 사람이 사전에 루브릭을 정의하는 대신, 2.2절에서 언급된 SPUR 방법론과 유사하게 LLM이 스스로 루브릭을 학습하고 도출하는 방식을 선택한다. 이를 위해 세션의 효율성을 확인할 수 있는 목적 지표들이 함께 주어진 AI 코딩 에이전트 채팅 히스토리 데이터셋을 활용한다. (본 논문에서는 토큰 효율성, 사용자 피드백 효율성 두 가지 목적 지표를 판단하는 루브릭을 도출한다). 지도 학습 파이프라인은 이 데이터셋을 활용하여, 각 판단 지표 별 핵심적인 특징(feature)들을 스스로 학습하여 이를 평가 루브릭으로 생성한다. 예를 들어, LLM은 높은 토큰 활용도의 세션에서 공통적으로 발견되는 ’명확한 요구사항 제시’, ’충분한 맥락 정보 제공’, ’AI 제안에 대한 비판적 피드백’ 등의 패턴을 학습하여, 이를 ’요구사항의 명확성 (Clarity of Prompt)’, ’맥락 제공 (Context Provision)’, ’도구 효율적 사용 (Efficient Tool Utilization)’ 등의 평가 루브릭으로 생성한다.
2. **루브릭 기반의 LLM-as-a-judge 평가:** LLM에 의해 자동 도출된 루브릭을 기반으로, LLM (예: gemini-2.5-flash [4])이 ’평가자(Judge)’ 역할을 수행하도록 하는 프롬프트를 설계한다. 이 프롬프트는 LLM에게 특정 채팅 히스토리 세션을 입력하여, 학습된 각 루브릭 항목별로 1~5점 척도의 점수를 매기고 그 근거를 서술하도록 지시한다.
3. **점수화 시스템 구축:** 2단계에서 LLM-as-a-judge 방법으로 도출한 각 루브릭별 점수(1~5점 척도)는 사용자의 AI 활용 능력을 세부적으로 진단하는 개별 지표로 활용된다. 또한 이 개별 루브릭 점수들의 산술 평균(arithmetic mean)을 계산하여 목적 지표에 대한 예측의 정확성을 검증하는 수단으로도 활용한다. 이 방식은 사용자에게 자신의 강점과 약점이 어떤 루브릭 항목에 있는지 직관적으로 파악하게 함과 동시에, 전반적인 활용 수준을 하나의 점수로 요약하여 검증의 신뢰도를 높이는 데 도움을 준다.

²<https://github.com/sanggggg/retrochat-evaluatator>

3.2.2 지도 학습 기반의 AI 활용 능력 루브릭(Rubric) 자동 도출: 훈련 파이프라인 구현 세부

진행한 훈련의 목적은 정량적 톤과 효율, 사용자 텐 흐름 등 세션의 효율성을 확인할 수 있는 지표들을 높은 정확도로 예측해 낼 수 있는 루브릭을 자동으로 도출하는 것이다.

훈련 파이프라인은 대략 다음과 같다. 훈련 데이터 셋에서 목적 지표가 높은 상위 퍼센타일 (상위 15%) 세션을 선별한 후, 각 세션을 구조화된 대화 세션 JSON 으로 변환한 뒤 루브릭 추출 프롬프트 템플릿을 사용하여 LLM 으로 루브릭을 추출한다. 이후 각 세션 별로 만들어진 3~5 개의 루브릭 들을 두 가지 요약 전략(LLM 요약 프롬프트 템플릿을 통한 요약 또는 텍스트 임베딩을 통한 의미론적 클러스터링)을 통해 최종 루브릭을 생성한다. 세부 단계는 다음과 같다

세션 로딩 및 필터링 데이터셋에서 점수를 기반으로 상위 15% 퍼센타일을 계산하고, 학습/검증 분리를 유지한 채 필요한 세션만 불러온다. 이를 통해 대규모 로그 중에서도 신뢰도 높은 세션만 선별적으로 학습에 투입한다.

루브릭 추출 루브릭 추출기는 각 세션을 JSON 문맥으로 포맷하고, 최소·최대 루브릭 개수를 명시해 Gemini 2.5 Flash 모델과 별도의 루브릭 추출 스크립트로 루브릭을 추출한다. 추출 결과는 JSON 배열 형태로 세션당 0~5개의 루브릭을 확보한다. 루브릭 추출 스크립트의 전체 구현 및 프롬프트 템플릿은 부록 5.1에 수록하였다.

루브릭 통합 수집된 수십~수백 개의 루브릭은 두 가지 전략으로 정제된다. 기본 전략은 요약 전용 프롬프트를 사용해 3~10개의 대표 기준으로 재구성하는 방식이며, context window가 과도하게 큰 경우 SPUR와 유사한 형태로 재귀적으로 100개 단위로 분할하여 summarize를 수행한다. 다른 전략은 Google 텍스트 임베딩 모델 gemini-embedding-001 [5] 과 UMAP + HDBSCAN 을 결합한 의미론적 클러스터링을 적용해 유사 루브릭을 자동 병합한다. UMAP을 통한 차원 축소와 HDBSCAN 클러스터링의 조합은 밀도에 따른 유연한 군집화와 고차원 임베딩 데이터에 대한 원활한 클러스터링을 가능하게 한다 [6]. 군집 크기는 패턴의 빈도를 의미하므로 사용자의 공통 행동 양상이 자연스럽게 강조된다. 요약 전용 프롬프트를 통한 루브릭 통합 과정의 세부 구현 및 실제 프롬프트는 부록 5.2에 추가로 수록하였다.

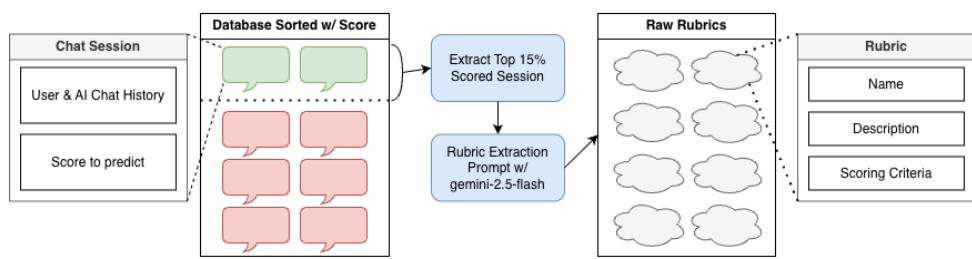


Figure 3.2: 지도 학습 기반의 Rubric 자동 도출 과정

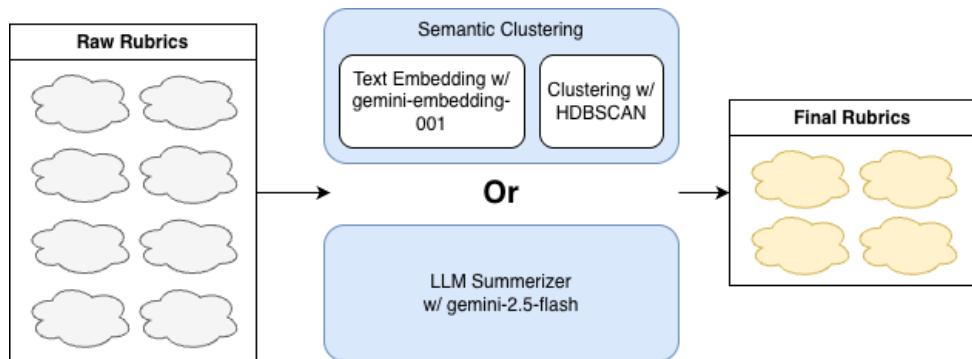


Figure 3.3: 루브릭 통합 과정 (LLM 요약 및 의미론적 클러스터링)

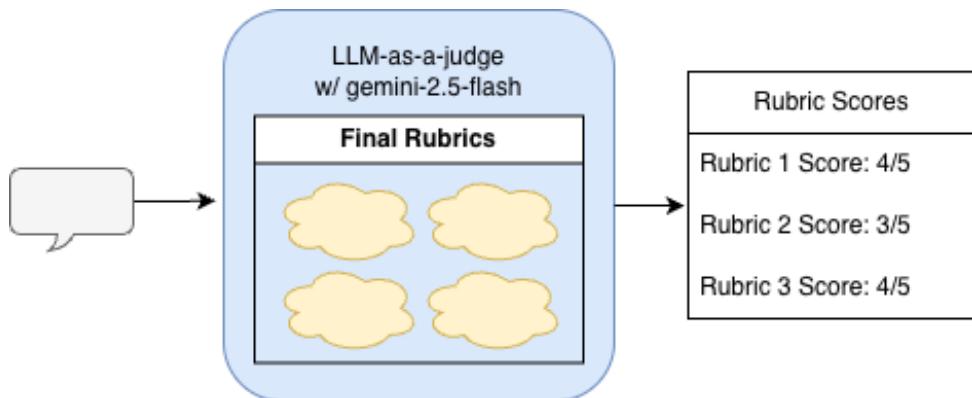


Figure 3.4: 학습된 Rubric 기반의 채팅 평가

4. 평가

4.1 평가 및 학습용 데이터셋 구축

4.1.1 원천 데이터 구성

RetroChat-Evaluator 저장소에는 매니페스트 생성 스크립트가 포함되어 있으며, Claude Code JSONL 로그를 재귀적으로 훑어 세션별 메타데이터와 점수를 생성한다. 실험에 사용된 원천 데이터는 10명의 Vibe Coders 를 통해 확보한 1,000개의 Claude Code 세션으로 구성되어 있으며, 모든 세션이 사용자 로컬에서 추출한 JSONL 파일 형태로 제공된다. 이 스크립트는 각 파일의 토큰 사용량, 도구 호출 횟수, 사용자 턴 수, 코드 변경(편집/작성 지시)에서 추정한 순증 LOC 등을 계산하고 토큰 효율(순증 LOC 대비 사용 토큰), 사용자 턴 효율(순증 LOC 대비 사용자 턴) 스코어를 함께 기록한다.

4.1.2 세션별 정답 스코어 정의

세션의 “정답” 스코어는 루브릭 학습 및 검증 단계에서 기준이 되는 정량 지표다. 매니페스트 생성 시 다음과 같이 계산된다.

- **토큰 효율** = 순증 LOC / 총 토큰 수. Claude Code가 생성한 최종 코드 라인 변화 대비 투입 토큰 수를 측정한다.
- **사용자 턴 효율** = 순증 LOC / 사용자 턴 수. 사용자 발화 한 번당 코드 성장량을 나타낸다.

이들 스코어는 데이터 로더를 통해 훈련·검증 파이프라인에 전달되며, 학습 시에는 상위 퍼센타일 필터 기준으로 동작하고, 검증 시에는 LLM이 예측한 총점과의 상관·오차를 계산하는 기준 값으로 사용된다.

4.2 지도학습 결과

본 연구에서는 토큰 효율(token_efficiency)과 사용자 턴 효율(user_turn_efficiency)을 목적 점수로 설정하고, 각각에 대해 상위 퍼센타일 세션으로부터 루브릭을 추출하였다. 루브릭 생성

방식으로는 LLM 기반 요약과 UMAP + HDBSCAN 기반 의미론적 군집화를 각각 적용하여 그 결과를 비교하였으며, 구체적인 루브릭 전문은 부록 5.3 및 5.4에 수록되어 있다.

4.2.1 토큰 효율 기준 루브릭

LLM 기반 요약 방식

상위 15 퍼센타일에 해당하는 70개 세션에서 277개의 개별 루브릭을 추출하였으며, LLM이 이들을 의미적으로 유사한 항목끼리 병합하도록 프롬프트를 구성한 결과, 최종 3개의 대표 루브릭이 도출되었다. 주요 루브릭으로는 초기 요청의 명확성과 완결성(Clarity and Completeness of Initial Request), 효율적이고 실행 가능한 반복(Efficient and Actionable Iteration), 전략적 위임과 자율성 부여(Strategic Delegation and Autonomy Enablement) 등이 포함되었다. 이들 루브릭은 사용자 행동의 초기 설정, 진행 중 안내, 작업 위임이라는 세 가지 핵심 단계를 포괄한다.

HDBSCAN 기반 군집화 방식

동일한 277개의 루브릭을 UMAP(n_neighbors=15, n_components=5, metric=cosine)으로 차원 축소 후 HDBSCAN(min_cluster_size=2)으로 군집화한 결과, 총 39개의 클러스터가 형성되었다. 이 중 가장 큰 5개의 클러스터를 선택하였으며, 군집 크기 분포는 [16, 12, 11, 10, 9]로 나타났다. 최종 5개의 루브릭은 상위 레벨 작업 위임(High-Level Task Delegation), 묶음 다단계 지시 (Bundled Multi-Step Instructions), 간결하고 구체적인 문제 보고(Concise and Specific Problem Reporting), 사전 설계 가이드 제시(Proactive Design Guidance), AI 프로세스에 대한 최소 개입과 신뢰(Minimal Interruption & Trust in AI's Process)로 구성되었다.

HDBSCAN 방식은 의미적 밀도가 높은 항목을 중심으로 클러스터를 형성하며, LLM 요약 방식 보다 더 세분화된 루브릭을 생성하는 경향을 보였다. 이는 임베딩 공간에서의 밀도 기반 군집화가 유사한 패턴을 자동으로 그룹화하면서도, LLM 요약이 통합한 일부 상위 개념을 더 구체적인 하위 패턴으로 유지하기 때문으로 해석된다.

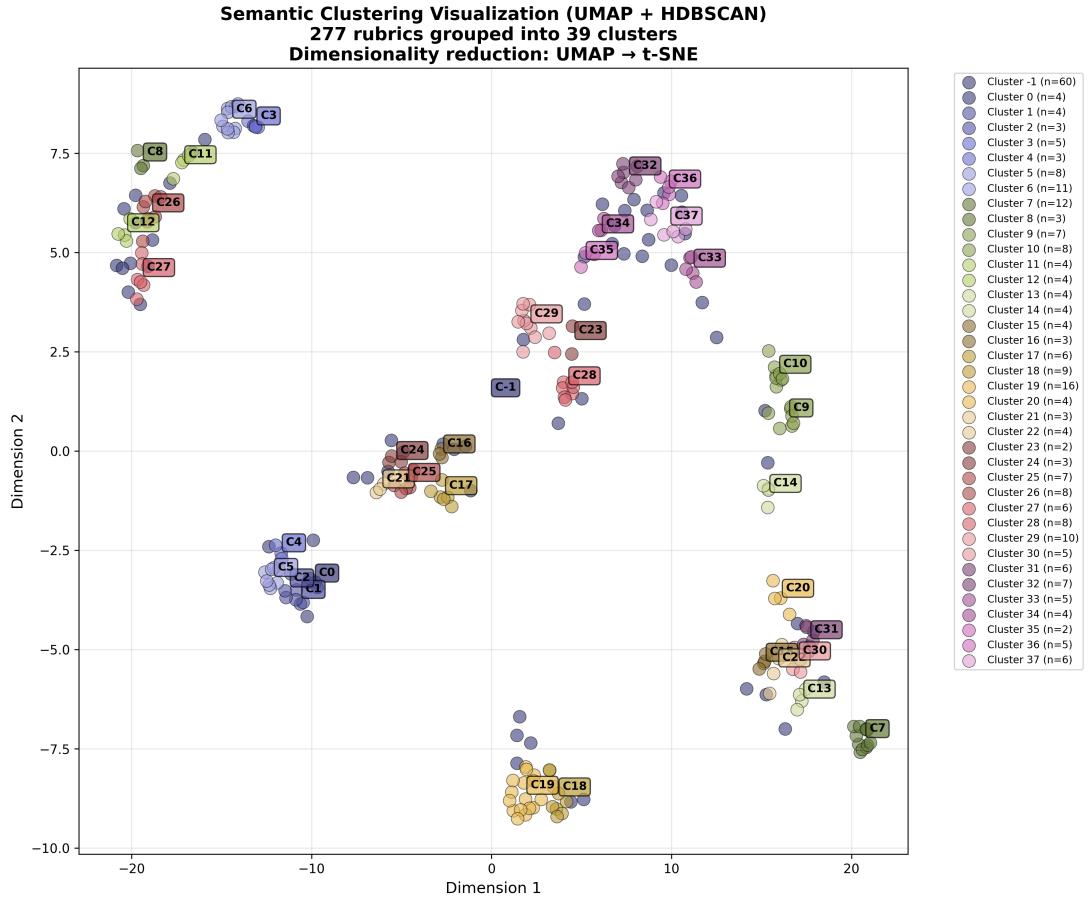


Figure 4.1: 토큰 효율 기준 루브릭의 UMAP + HDBSCAN 군집화 시각화. 277개의 루브릭이 임베딩 공간에서 39개의 클러스터로 그룹화되었다.

4.2.2 사용자 텐 효율 기준 루브릭

LLM 기반 요약 방식

사용자 텐 효율을 목적 점수로 설정한 경우, 상위 15 퍼센타일에 해당하는 70개 세션에서 274 개의 루브릭을 추출하였으며, LLM 요약 결과 3개의 루브릭이 도출되었다. 대표적으로 **명확하고 포괄적인 초기 요청(Clear and Comprehensive Initial Request)**, **실행 가능한 반복적 안내와 피드백(Actionable Iterative Guidance and Feedback)**, **전략적 위임과 AI 자율성에 대한 신뢰(Strategic Delegation and Trust in AI Autonomy)** 등이 포함되었다. 이들 루브릭은 토큰 효율 루브릭과 유사하게 초기 요청, 진행 중 피드백, 작업 위임이라는 세 단계를 다루지만, 대화 텐 자체의 품질과 효율성을 더 강조하는 특징을 보인다.

HDBSCAN 기반 군집화 방식

동일한 274개의 루브릭을 UMAP(n_neighbors=15, n_components=5, metric='cosine')으로 차원 축소 후 HDBSCAN(min_cluster_size=2)으로 군집화한 결과, 총 36개의 클러스터가 형성되었다. 이 중 가장 큰 5개의 클러스터를 선택하였으며, 군집 크기 분포는 [17, 14, 14, 12, 10]으로 나타났다. 최종 5개의 대표 루브릭은 다음과 같다.

- **정확한 문제 식별(Precise Problem Identification):** 이슈나 버그를 보고할 때 정확한 문제 출력이나 구체적 맥락을 제공하여 AI가 신속히 근본 원인을 파악하도록 하는 능력을 평가한다.
- **AI의 계획과 실행에 대한 신뢰(Trust in AI's Planning and Execution):** AI가 제공된 요구사항에 따라 자율적으로 계획하고 실행할 수 있도록 허용하는 사용자의 의지를 측정한다.
- **명확한 작업 지시(Clear Task Directives):** 필요한 맥락과 매개변수를 포함하여 AI가 즉시 행동할 수 있는 구체적이고 잘 정의된 작업을 지시하는 능력을 평가한다.
- **관련 지시 묶음(Bundling Related Instructions):** 논리적으로 연결된 지시사항이나 요구 사항을 단일 메시지로 그룹화하여 AI가 여러 측면을 동시에 처리하도록 하는 능력을 측정한다.
- **상위 레벨 목표 정의(High-Level Goal Definition):** 포괄적이고 복잡한 목표를 사전에 정의하여 AI가 다단계 작업을 자율적으로 계획하고 실행하도록 하는 능력을 평가한다.

사용자 턴 효율 루브릭은 토큰 효율 루브릭에 비해 대화 턴 최소화와 효율적 의사소통 패턴을 더 강조하는 경향을 보이며, 이는 두 목적 점수가 서로 다른 차원의 사용자 역량을 측정함을 시사한다.

4.3 검증 결과

Validation split 52개 세션에 대해 각 목적 점수 기준으로 생성된 루브릭을 적용하여 평가를 수행하였다. 루브릭 평가는 1~5점 스케일로 이루어지며, 목적 점수는 전체 데이터셋 내에서의 퍼센타일로 표현된다. 두 척도 간의 상관관계를 측정하기 위해 Kendall의 tau 상관계수 [7]를

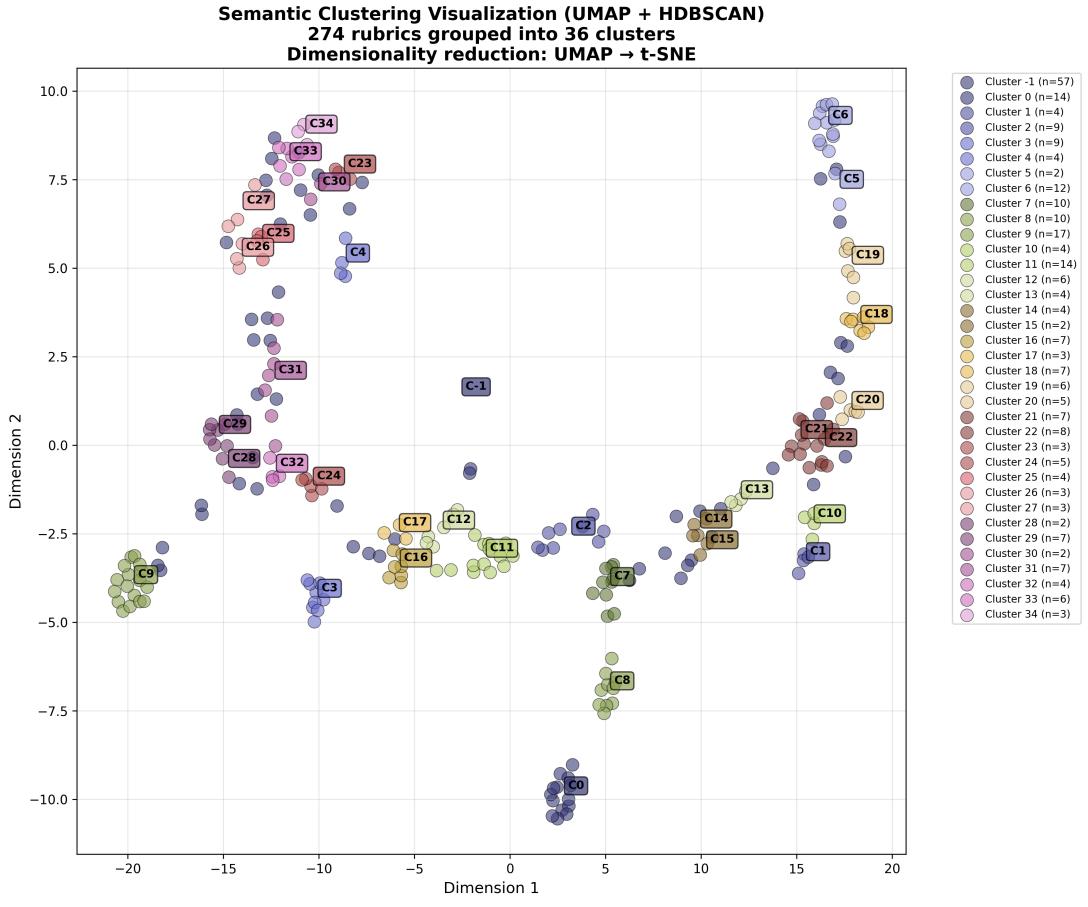


Figure 4.2: 사용자 턴 효율 기준 루브릭의 UMAP + HDBSCAN 군집화 시각화. 274개의 루브릭이 임베딩 공간에서 36개의 클러스터로 그룹화되었다.

사용하였다. Pearson 상관계수 대신 Kendall의 tau를 선택한 이유는 다음과 같다. 첫째, 루브릭 점수(1~5)와 목적 점수 퍼센타일(0~100)의 척도 범위가 상이하여 원점수 간 선형 관계를 직접 비교하기 어렵다. 둘째, 루브릭 점수는 이산적 서열 척도이고 목적 점수 퍼센타일은 비선형 분포를 따르므로, 두 변수 간의 단조 관계(monotonic relationship)를 측정하는 순위 기반 상관계수가 더 적합하다. 셋째, Kendall의 tau는 Spearman 상관계수에 비해 이상치에 더 강건하고, 소표본에서도 안정적인 추정치를 제공하며, 통계적 검정의 가정을 덜 요구한다는 장점이 있다 [8].

표 4.1는 두 가지 루브릭 생성 방식(CLUSTER, LLM)과 두 가지 목적 점수(토론 효율, 사용자 턴 효율)에 대한 검증 결과를 요약한 것이다.

모든 경우에서 p 값이 0.05를 초과하여 통계적으로 유의미한 상관관계가 발견되지 않았다. 토론 효율 기준에서는 CLUSTER 방식이 가장 높은 tau 값(0.1174)을 기록하였으나, 여전히 약한 상관에 그쳤다.

Table 4.1: 루브릭 생성 방식별 검증 결과

방식	목적 점수	Kendall's tau	p-value
CLUSTER	토큰 효율	0.1174	0.271
LLM	토큰 효율	0.0991	0.330
CLUSTER	사용자 턴 효율	0.0353	0.448
LLM	사용자 턴 효율	0.0456	0.355

4.3.1 토큰 효율 기준 검증 결과

LLM 요약 루브릭 평가

LLM이 생성한 3개의 루브릭을 적용한 결과, Kendall의 tau는 0.0991 ($p=0.330$)로 나타나 통계적으로 유의미하지 않은 약한 양의 상관을 보였다. p값이 0.05보다 크므로 귀무가설(두 변수 간 상관이 없음)을 기각할 수 없으며, 이는 LLM이 의미적으로 일관된 루브릭을 생성했음에도 불구하고, 루브릭 점수가 실제 목적 점수(토큰 효율)의 순위를 예측하는 데 한계가 있음을 시사한다.

HDBSCAN 군집화 루브릭 평가

HDBSCAN 기반 5개 루브릭을 적용한 결과, Kendall의 tau는 0.1174 ($p=0.271$)로 측정되었다. LLM 방식($\tau=0.0991$)보다 다소 높은 상관계수를 보였으나, p값이 0.271로 여전히 통계적 유의수준(0.05)을 초과하여 귀무가설을 기각할 수 없다. 이는 의미론적 군집화가 더 세분화된 패턴을 포착하여 LLM 요약 방식보다 약간 나은 예측 경향을 보이지만, 통계적으로 유의미한 예측력을 갖지는 못함을 의미한다. 전반적으로 루브릭만으로는 토큰 효율을 정확히 예측하기 어려움을 확인할 수 있다.

4.3.2 사용자 턴 효율 기준 검증 결과

LLM 요약 루브릭 평가

사용자 턴 효율 기준으로 LLM이 생성한 3개의 루브릭을 적용한 결과, Kendall의 tau는 0.0456 ($p=0.355$)로 나타났다. p값이 0.355로 통계적 유의수준을 크게 초과하여, 루브릭 점수와 사용자 턴 효율 간에 유의미한 상관관계가 없음을 보였다. 이는 사용자 턴 효율이 대화 전략과 의사소통 패턴을 직접 반영하는 지표임에도 불구하고, LLM이 생성한 질적 루브릭이 정량적 턴 효율

지표를 예측하는 데 한계가 있음을 시사한다.

HDBSCAN 군집화 루브릭 평가

HDBSCAN 기반 5개 루브릭을 적용한 결과, Kendall의 tau는 0.0353 ($p=0.448$)으로 측정되었다. LLM 방식($\tau=0.0456$)과 유사한 수준의 매우 약한 양의 상관을 보였으며, p 값이 0.448로 통계적 유의성이 전혀 없음을 확인하였다. 토큰 효율 기준과 달리 군집화 방식이 추가적인 예측 능력 향상을 제공하지 못했으며, 이는 사용자 턴 효율 관련 루브릭이 임베딩 공간에서 명확한 군집 구조를 형성하지 못했거나, 턴 효율이라는 정량 지표와 질적 루브릭 패턴 간의 연결이 토큰 효율보다 더 약함을 의미한다.

4.3.3 척도 간 상관관계 해석

루브릭 점수(1~5 스케일)와 목적 점수 퍼센타일(실수 범위) 간의 Kendall의 tau는 전반적으로 매우 낮은 수준($0.035 \sim 0.117$)을 보였으며, 모든 경우에서 통계적으로 유의미하지 않았다($p \leq 0.05$). 구체적으로, 토큰 효율 기준에서는 HDBSCAN 방식이 $\tau=0.1174$ ($p=0.271$), LLM 방식이 $\tau=0.0991$ ($p=0.330$)을 기록하였고, 사용자 턴 효율 기준에서는 LLM 방식이 $\tau=0.0456$ ($p=0.355$), HDBSCAN 방식이 $\tau=0.0353$ ($p=0.448$)을 기록하였다. 루브릭은 사용자 행태의 질적 패턴을 기반으로 하는 반면, 목적 점수는 코드 라인 수 변동 대비 토큰 사용량 또는 사용자 턴 수와 같은 정량 지표에 의존한다. 따라서 두 척도는 본질적으로 서로 다른 차원을 측정하며, 강한 단조 관계를 기대하기 어렵다. 토큰 효율 기준에서는 두 방식 모두 약한 양의 상관 경향을 보였으나, 사용자 턴 효율 기준에서는 상관이 거의 없는 것으로 나타났다. 이는 루브릭이 포착하는 질적 패턴이 정량적 효율 지표와 직접적인 인과관계를 갖지 않거나, 현재 목적 점수가 사용자 행태의 질을 충분히 반영하지 못함을 시사한다.

4.3.4 낮은 상관관계의 원인 분석 및 개선 방향

Kendall의 tau가 전반적으로 낮고 통계적으로 유의미하지 않게 나타난 이유는 다음과 같다. 첫째, 실측 스코어가 순증 LOC와 토큰/턴 수만을 반영하는 단순 지표인 반면, 루브릭은 목표 설정 명확성, 오류 처리 전략, 맵락 인식, 작업 위임 등 복합적 행태를 포착한다. 이 두 척도는 본질적으로 서로 다른 개념을 측정하며, 질적 패턴과 정량적 효율이 항상 일치하지는 않는다. 둘째,

코드 변경이 없거나 탐색 위주의 세션(순증 LOC=0)은 목적 점수가 0 또는 매우 낮게 책정되지만, 루브릭은 사용자의 질문 품질과 탐색 전략을 평가하여 상대적으로 높은 점수를 부여할 수 있다.셋째, 순증 LOC 자체가 불안정한 지표로, 동일한 품질의 코드 작성이라도 프로젝트 특성(보일러플레이트 코드 유무, 리팩토링 여부 등)에 따라 큰 편차를 보인다. 넷째, 검증용 표본 크기(52개 세션)가 제한적이어서 통계적 검정력이 낮아졌을 가능성이 있다.

이러한 간극을 줄이기 위한 개선 방향으로는 다음과 같은 접근이 필요하다. 첫째, 코드 실행 성공률, 테스트 통과율, 빌드 성공 여부 등 결과 기반 지표를 추가 목적 점수로 통합한다. 둘째, 사람이 직접 부여한 세션 품질 레이블을 수집하여 루브릭 학습의 기준으로 활용한다. 셋째, 루브릭 자체를 목적 점수 예측이 아닌 사용자 행태의 질적 평가 도구로 활용하고, 피드백 제공 목적으로 재정립한다. 이는 향후 과제로 남겨둔다.

5. 결론

본 연구는 AI 코딩 에이전트와의 상호작용 데이터를 다면적으로 수집하는 RetroChat 툴킷과, 수집된 세션으로부터 사용자 숙련도 루브릭을 자동 학습·적용하는 LLM-as-a-judge 파이프라인을 구현하였다. Trainer는 상위 퍼센타일 세션만을 선별해 비동기 LLM 호출로 루브릭을 추출하고, LLM 요약 또는 의미론적 클러스터링을 통해 대표 루브릭을 생성한다. Evaluator는 동일 루브릭을 이용해 신규 세션을 1~5점 척도로 점수화하고, 결과를 JSON으로 저장함으로써 분석 자동화를 달성하였다.

실험적으로는 10명의 Vibe Coders로부터 확보한 1,000개 Claude Code 세션을 기반으로, 토큰 효율과 사용자 턴 효율이라는 두 가지 목적 점수에 대해 각각 루브릭을 생성하였다. 토큰 효율 기준으로는 상위 15%에 해당하는 70개 세션에서 277개 루브릭을 추출하였으며, LLM 기반 요약 방식으로 3개의 대표 루브릭을, UMAP + HDBSCAN 기반 군집화 방식으로는 98개 클러스터 중 상위 5개를 선택하여 5개의 대표 루브릭을 도출하였다. 사용자 턴 효율 기준으로는 70개 세션에서 274개 루브릭을 추출하였으며, LLM 방식으로 3개, HDBSCAN 방식으로 5개의 대표 루브릭을 생성하였다. 52개 세션으로 구성된 검증 데이터셋에 대해 Kendall의 tau 상관계수를 측정한 결과, 토큰 효율 기준에서는 HDBSCAN 방식이 $\tau = 0.1174$ ($p = 0.271$), LLM 방식이 $\tau = 0.0991$ ($p = 0.330$)을, 사용자 턴 효율 기준에서는 LLM 방식이 $\tau = 0.0456$ ($p = 0.355$), HDBSCAN 방식이 $\tau = 0.0353$ ($p = 0.448$)을 기록하였다. 모든 경우에서 p 값이 0.05를 초과하여 통계적으로 유의미한 상관관계가 발견되지 않았으나, 토큰 효율 기준의 HDBSCAN 방식이 상대적으로 가장 높은 상관 경향을 보였다.

그러나 데이터의 정답 라벨이 순증 LOC 대비 토큰/턴 수라는 단순 정량 지표에 치우쳐 있고, 모델이 참조할 추가 맥락(예: 빌드 성공 여부, 테스트 통과율, 코드 리뷰 피드백)이 부족하다는 점이 한계로 남는다. 또한 질적 루브릭과 정량 목적 점수 간의 본질적 차이로 인해 두 척도가 서로 다른 차원을 측정하며, 검증 표본 크기(52개)가 제한적이어서 통계적 검정력이 낮아진 것으로 분석된다.

이러한 한계에도 불구하고 본 연구의 루브릭 기반 평가 방식은 중요한 시사점을 제공한다. 기준의 단순 정량 지표(토큰 사용량, 대화 턴 수 등)는 사용자에게 “얼마나” 효율적이었는지만 알려줄 뿐, “어떻게” 개선해야 하는지에 대한 구체적 방향을 제시하지 못한다. 반면 본 연구의 루브릭은 초기 요청의 명확성, 반복적 피드백의 실행 가능성, 작업 위임과 자율성 부여 등 사용

자 행동의 특정 측면을 세분화하여 평가하고, 각 항목별로 1~5점 척도의 점수와 함께 구체적인 개선 근거를 제공한다. 이는 사용자가 자신의 상호작용 패턴에서 어떤 부분이 우수하고 어떤 부분이 미흡한지를 이해할 수 있게 하며, 설명 가능한 피드백(explainable feedback)을 통해 AI 코딩 에이전트 활용 능력의 체계적 개선을 가능하게 한다. 특히 초보 사용자의 경우, 상위 퍼센타일 세션으로부터 학습된 루브릭을 참고하여 효과적인 프롬프팅 전략과 작업 위임 방식을 학습할 수 있으며, 이는 단순한 점수 비교를 넘어 실질적인 역량 향상으로 이어질 수 있다.

이러한 설명 가능한 피드백은 본 연구의 또 다른 핵심 기여인 'RetroChat' GUI 애플리케이션을 통해 사용자에게 전달된다. RetroChat은 데이터 수집 도구를 넘어, 루브릭 기반 평가 결과를 직관적으로 시각화하고 사용자의 성장을 돋는 실용적인 분석 및 회고 플랫폼으로서 기능한다. 사용자는 자신의 과거 채팅 세션을 단순히 다시 읽는 것을 넘어, 각 세션이 어떤 기준으로 평가되었는지, 자신의 어떤 상호작용 방식이 긍정적 혹은 부정적 평가를 받았는지에 대한 상세한 피드백을 GUI를 통해 직접 확인할 수 있다. 이는 복잡한 LLM-as-a-judge 파이프라인의 분석 결과를 최종 사용자가 쉽게 소비하고 자신의 AI 활용 능력을 성찰하는 도구로 사용할 수 있게 만든다는 점에서 큰 의의를 가진다.

향후에는 (1) RetroChat 툴킷을 통한 더 다양한 벤더/언어 세션 확보, (2) 도메인 전문가의 직접 어노테이션을 포함한 다중 레이블 구축, (3) 코드 실행 성공률, 테스트 통과율 등 결과 기반 지표의 통합, (4) 루브릭 가중치 최적화 및 메타-평가(ensemble judges) 도입 등을 통해 보다 신뢰도 높은 사용자 숙련도 평가 지표를 완성하고자 한다.

참고문헌

- [1] ryoppippi, “ccusage,” <https://github.com/ryoppippi/ccusage>, n.d., GitHub repository.
- [2] sculptdotfun, “viberank,” <https://github.com/sculptdotfun/viberank>, n.d., GitHub repository.
- [3] Y.-C. Lin, J. Neville, J. W. Stokes, L. Yang, T. Safavi, M. Wan, S. Counts, S. Suri, R. Andersen, X. Xu, D. Gupta, S. K. Jauhar, X. Song, G. Buscher, S. Tiwary, B. Hecht, and J. Teevan, “Interpretable user satisfaction estimation for conversational systems with large language models,” 2024. [Online]. Available: <https://arxiv.org/abs/2403.12388>
- [4] Gemini Team, Google, “Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities,” *arXiv preprint arXiv:2507.06261*, 2025. [Online]. Available: <https://arxiv.org/abs/2507.06261>
- [5] ——, “Gemini embedding: Generalizable embeddings from gemini,” *arXiv preprint arXiv:2503.07891*, 2025. [Online]. Available: <https://arxiv.org/abs/2503.07891>
- [6] R. J. G. B. Campello, D. Moulavi, and J. Sander, “Density-based clustering based on hierarchical density estimates,” in *Advances in Knowledge Discovery and Data Mining (PAKDD 2013)*, ser. Lecture Notes in Computer Science, J. Pei, V. S. Tseng, L. Cao, H. Motoda, and G. Xu, Eds., vol. 7819. Springer, Berlin, Heidelberg, 2013, pp. 160–172.
- [7] M. G. Kendall, “A new measure of rank correlation,” *Biometrika*, vol. 30, no. 1/2, pp. 81–93, 1938.
- [8] C. Croux and C. Dehon, “Influence functions of the spearman and kendall correlation measures,” *Stat methods Appl*, vol. 19, no. 4, pp. 497–515, 2010.

부록

5.1 Rubric Extraction Prompt Template

Rubric Extraction Prompt

You are an expert at analyzing AI agent interactions and identifying patterns that indicate high-quality user behavior.

Analyze the following chat session between a human user (messages with `role: user`) and an AI coding assistant. This session has been rated as high-quality. All rubrics you extract must describe only the human user's behavior—never the AI assistant's actions.

Chat Session

{chat_session}

Task

Extract a list of evaluation rubrics that capture what makes this user interaction effective.

Focus on:

- How clearly the user communicated their requirements
- How efficiently the user guided the AI toward the solution
- How well the user handled clarifications and corrections
- Any patterns of effective AI agent usage

Output Format

Provide your response as a JSON array of rubrics. Each rubric should have:

- name: Short descriptive name (2-5 words)
- description: What this rubric measures (1-2 sentences)
- scoring_criteria: How to score from 1 (poor) to 5 (excellent)
- evidence: Specific example from this session demonstrating the rubric

Extract {extraction_min_rubrics}-{extraction_max_rubrics} rubrics that are specific and actionable.

5.2 Rubric Summarization Prompt Template

Rubric Summarization Prompt

You are an expert at synthesizing evaluation criteria for AI agent interactions.

You have been given rubrics extracted from multiple high-quality chat sessions. Your task is to consolidate these into a final, coherent set of evaluation rubrics that assess only the human user (messages labeled `role: user`) and never the AI assistant.

Extracted Rubrics from All Sessions

{all_rubrics}

Task

Create a final list of {summarization_min_rubrics}-{summarization_max_rubrics} evaluation rubrics by:

1. Identifying common themes across the extracted rubrics
2. Merging similar or overlapping rubrics
3. Removing redundant or overly specific rubrics
4. Ensuring comprehensive coverage of user efficiency aspects
5. Making criteria clear and consistently scorable

Output Format

Provide your response as a JSON object with the final rubrics. Ensure each rubric:

- Has a unique, descriptive name
- Has clear, objective scoring criteria
- Is applicable across different types of coding tasks
- Focuses exclusively on the human USER (messages with `role: user`), not AI performance

5.3 Token Efficiency Rubrics

5.3.1 LLM Summarization Method (3 rubrics)

1. Clarity and Completeness of Initial Request

Description: The user provides a complete, clear, and well-structured initial request, including all necessary context, goals, constraints, and desired output formats, enabling the AI to immediately understand the task and formulate an actionable plan without extensive clarification.

Scoring Criteria:

- 1: Vague, ambiguous, or highly incomplete initial request, requiring multiple turns for basic understanding and task setup.
- 2: Provides basic requirements but lacks significant context, constraints, or specific details, leading to moderate clarification.
- 3: Sets a clear goal and provides some context, but still requires the AI to infer details or ask for minor clarifications on constraints or output.
- 4: Delivers clear and comprehensive initial requirements, with most essential information provided upfront, allowing the AI to form a good plan.
- 5: Consistently provides highly detailed, structured, and self-contained initial requests that anticipate potential issues and constraints, enabling the AI to autonomously plan and execute complex tasks with minimal or no initial clarification.

2. Efficient and Actionable Iteration

Description: The user provides concise, timely, and actionable feedback, corrections, and follow-up instructions, leveraging AI's context and previous output, bundling related requests, and efficiently guiding the AI's iterative process with minimal conversational overhead. This includes precise and timely course correction.

Scoring Criteria:

- 1: Feedback is vague, delayed, fragmented, or redundant; requires extensive AI clarification or re-states information already known.
- 2: Provides basic feedback or answers, but often requires follow-ups, lacks specificity, or fails to leverage AI's context effectively.
- 3: Delivers generally clear and timely feedback/answers, but could be more concise, proactive in bundling instructions, or precise in error correction.
- 4: Consistently provides clear, specific, and timely feedback, often bundling related instructions or corrections, and leverages AI's context effectively.
- 5: Provides immediate, precise, and minimal feedback, corrections, and consolidated answers to AI queries, directly leveraging AI's previous output and context

to efficiently steer the conversation and rectify errors with minimal token waste.

3. Strategic Delegation and Autonomy Enablement

Description: The user effectively delegates complex, multi-step tasks at a high conceptual level, provides strategic input, and allows the AI sufficient autonomy to plan and execute, minimizing micro-management and maximizing AI's analytical and generative potential.

Scoring Criteria:

- 1: Micro-manages the AI, breaks down complex tasks into overly granular steps, or frequently interrupts AI's workflow, failing to leverage its autonomous capabilities.
- 2: Delegates simple tasks but struggles with complex ones, or intervenes frequently with unnecessary checks, not fully trusting AI's autonomy.
- 3: Delegates moderately complex tasks and provides some strategic input, but may occasionally interrupt AI's autonomy or miss opportunities for deeper leveraging.
- 4: Effectively delegates complex tasks, provides relevant strategic guidance (e.g., external references, design patterns), and generally allows the AI to execute autonomously.
- 5: Consistently delegates highly complex, multi-faceted tasks with clear output specifications, proactively provides strategic context and design principles, and trusts the AI to autonomously plan and execute, maximizing AI's analytical and generative potential.

5.3.2 HDBSCAN Clustering Method (5 rubrics)

1. High-Level Task Delegation

Description: Measures how effectively the user delegates large, multi-step tasks to the AI, relying on the AI's ability to interpret and execute based on existing context (e.g., a plan document). This minimizes the need for the user to break down complex tasks into granular steps.

Scoring Criteria:

- 1: User breaks down every step, requiring explicit instructions for each.
- 3: User delegates moderate tasks but still provides significant detail.
- 5: User delegates entire phases or large features with minimal explicit instruction, trusting the AI to manage sub-tasks.

2. Bundled Multi-Step Instructions

Description: Evaluates the user's ability to combine multiple related, sequential, or parallel tasks into a single, coherent prompt, minimizing turn overhead and allowing the AI to plan and execute a sequence of actions efficiently.

Scoring Criteria:

- 1: Each action is requested in a separate turn, leading to excessive back-and-forth.
- 3: Some related actions are grouped, but complex sequences are still broken down.
- 5: User effectively bundles several distinct but related instructions into one prompt, often including context or examples for each sub-task, enabling the AI to perform a complex operation without further prompting.

3. Concise and Specific Problem Reporting

Description: The user provides direct and unambiguous descriptions of observed problems or error messages, allowing the AI to quickly identify the root cause without requesting further diagnostic information.

Scoring Criteria:

- 1: Vague or incomplete problem descriptions requiring extensive AI clarification.
- 3: Adequate problem descriptions.
- 5: Precise error messages or behavioral descriptions that immediately guide the AI to a solution.

4. Proactive Design Guidance

Description: Evaluates the user's ability to provide forward-looking context or requirements that influence the AI's design choices for reusability or extensibility, preventing future refactoring.

Scoring Criteria:

- 1: User only addresses immediate needs, leading to potential rework for future extensions.
- 3: User hints at future needs but doesn't fully articulate them.
- 5: User explicitly states future use cases or design principles, allowing the AI to create a more robust and adaptable solution from the start.

5. Minimal Interruption & Trust in AI's Process

Description: The user allows the AI to execute its planned steps without unnecessary interruptions, redundant checks, or premature requests for status updates, demonstrating trust in the AI's ability to follow through.

Scoring Criteria:

- 1: Frequent interruptions, asking for status updates, or re-stating instructions already given.
- 3: Occasional interruptions or minor redundant checks.
- 5: User intervenes only when necessary (e.g., to correct a mistake or add a new task), allowing the AI to work through its plan efficiently.

5.4 User Turn Efficiency Rubrics

5.4.1 LLM Summarization Method (3 rubrics)

1. Clear and Comprehensive Initial Request

Description: Evaluates the user's ability to provide a complete, unambiguous, and actionable initial request, including all necessary context, goals, constraints, and desired output structure, enabling the AI to form a robust plan and begin substantial work without immediate clarification.

Scoring Criteria:

- 1: Provides minimal or vague information, requiring extensive clarification (multiple turns) for the AI to understand the basic task, context, or desired output.
- 2: Provides some basic information, but critical context, constraints, or output details are missing, leading to several clarification turns.

- 3: Provides most key details, but some ambiguities or minor gaps remain, necessitating a few clarification turns or assumptions by the AI.
- 4: Provides a thorough initial request, including clear goals, relevant context, and most constraints, allowing the AI to proceed with minimal or no clarification.
- 5: Provides an exceptionally detailed, well-structured, and unambiguous initial request, covering all essential context, specific goals, critical constraints, and desired output format, enabling the AI to autonomously generate a comprehensive plan or significant output immediately.

2. Actionable Iterative Guidance and Feedback

Description: Assesses the user's ability to provide clear, concise, and consolidated feedback, corrections, and new instructions during the interaction. This includes comprehensively answering AI's clarifying questions, providing specific diagnostic information, and maintaining an efficient conversation flow.

Scoring Criteria:

- 1: Provides vague, fragmented, or overly emotional feedback/instructions, requiring multiple turns for the AI to understand the problem or desired change. Answers AI questions individually or incompletely, or introduces new, unrequested topics.
- 2: Identifies issues but lacks specificity or bundles unrelated points, leading to minor AI confusion or requiring follow-up questions. Answers to AI questions are often incomplete or verbose.
- 3: Provides generally clear feedback/instructions and answers most AI questions in one turn, but might miss some details, opportunities to bundle related points, or consistently leverage implicit context.
- 4: Consistently provides clear, specific, and timely feedback/instructions, often bundling related points. Answers AI questions comprehensively in single turns, allowing immediate AI action.
- 5: Provides exceptionally precise, concise, and consolidated feedback, corrections, or new instructions, often including rationale, specific examples, or actionable diagnostic data. Responds to all AI clarifying questions completely and unambigu-

ously in a single turn, leveraging implicit context, and enabling immediate and accurate AI action.

3. Strategic Delegation and Trust in AI Autonomy

Description: Evaluates the user's skill in delegating complex tasks at a high level, providing sufficient information for the AI to plan and execute autonomously, and refraining from micro-management. This includes proactive problem framing, solution suggestion, and challenging AI assumptions to achieve optimal solutions.

Scoring Criteria:

- 1: Micro-manages the AI, providing step-by-step instructions for every small action, or frequently interrupts the AI's workflow, preventing autonomous progress. Fails to proactively frame problems or suggest solutions.
- 2: Delegates some sub-tasks but frequently intervenes with detailed instructions or checks in often, slowing down autonomous execution. Provides vague problem framing or unhelpful suggestions.
- 3: Allows the AI some autonomy for sub-tasks but might still provide unnecessary guidance or check-ins during execution. Frames problems with basic context or offers general solution ideas.
- 4: Delegates significant multi-step tasks at a high level, allowing the AI to plan and execute without constant intervention. Proactively frames problems with relevant context or offers specific, relevant hypotheses.
- 5: Consistently delegates major phases of a project with clear, high-level directives, trusting the AI to autonomously plan, execute, and problem-solve over many turns without interruption. Proactively provides precise technical hypotheses, solution suggestions, or authoritative references, significantly accelerating problem resolution and guiding towards architecturally sound solutions.

5.4.2 HDBSCAN Clustering Method (5 rubrics)

1. Precise Problem Identification

Description: When reporting an issue or bug, the user provides exact problematic out-

put or specific context, enabling the AI to quickly locate and address the root cause without extensive diagnostic queries.

Scoring Criteria:

- 1: User describes a problem vaguely, requiring the AI to ask many clarifying questions.
- 2: User provides a general description, but the AI struggles to pinpoint the exact issue.
- 3: User describes the problem with some detail, allowing the AI to narrow down the search.
- 4: User provides specific symptoms or partial output, guiding the AI effectively.
- 5: User provides the exact problematic output or code snippet, allowing the AI to directly search for and resolve the issue.

2. Trust in AI's Planning and Execution

Description: Assesses the user's willingness to allow the AI to autonomously plan and execute tasks based on the provided requirements, without excessive interruptions, micro-management, or requests for intermediate updates.

Scoring Criteria:

- 1: User constantly interrupts, micro-manages, or demands frequent updates, hindering AI's flow.
- 3: User provides some space but still frequently checks in or asks for minor adjustments during execution.
- 5: User provides clear instructions and then allows the AI to proceed with its plan, only intervening for significant new requirements or when prompted by the AI.

3. Clear Task Directives

Description: Measures how effectively the user directs the AI to a specific, well-defined task, including necessary context and parameters, to enable immediate action without requiring the AI to ask "what next?".

Scoring Criteria:

- 1: Vague or ambiguous instructions that require multiple clarification rounds.
- 3: Clear but requires minor AI clarification or prioritization.
- 5: Precise, self-contained directives that allow the AI to execute complex tasks without further user input.

4. Bundling Related Instructions

Description: This rubric measures how well the user groups logically connected instructions or requirements into a single message, allowing the AI to process and act on multiple aspects of the task simultaneously rather than in fragmented exchanges.

Scoring Criteria:

- 1: User provides instructions one at a time, even when related.
- 3: User occasionally bundles instructions but often separates them.
- 5: User consistently groups related requirements, clarifications, or modifications into single, coherent messages, maximizing the AI's ability to make progress in one turn.

5. High-Level Goal Definition

Description: Measures how effectively the user defines a broad, complex objective up-front, enabling the AI to plan and execute multi-step tasks autonomously without requiring detailed instructions for each sub-step.

Scoring Criteria:

- 1: User provides fragmented, unclear, or overly narrow initial requests, requiring extensive clarification.
- 3: User provides a clear goal but might miss some key aspects, leading to moderate clarification.
- 5: User articulates a comprehensive, high-level goal that allows the AI to take ownership of planning and execution without further input on the overall objective.