

# Agent 기반의 수업 보조 챗봇 시스템

신다연<sup>0</sup>, 이동훈  
고려대학교 수학과  
{yeon\_25, holy}@korea.ac.kr

## Agent-based Instructional Support Chatbot

Dayeon Shin<sup>0</sup>, Donghun Lee  
Department of Mathematics, Korea University

### 요약

기존 출석 방식들은 시간이 많이 소요되고, 부정행위의 가능성도 있다. 본 논문은 Zulip과 LangGraph를 기반으로 한 출석 자동화 시스템을 제안하며, 하이브리드 교육 환경에서 출결 관리의 효율성과 신뢰성 향상을 목표로 한다. LangGraph시스템에 기반하여 고정된 키워드 없이 학생의 자연어 응답을 해석하고 학생들의 출석을 DB에 기록한다. 약 350명을 대상으로 실제 수업에서 실험한 결과, 다양한 비정형 표현을 높은 정확도로 인식하였으며, 챗봇의 평균 응답 지연 시간은 3초 이내로 실시간 운영에 적합한 성능을 보였다. 실험 중 이메일 불일치나 반복 메시지 전송 등의 한계도 일부 발견되었으나, 향후 콘텐츠 필터링과 학습 피드백 기능 통합을 통해 연구 및 시스템의 확장 가능성을 제시한다.

### 1. 서론

최근 코로나19 팬데믹 등을 계기로 비대면 및 하이브리드 수업이 확산되면서 출결 관리의 중요성이 더욱 부각되고 있다. 그러나 현재 널리 사용되는 기존 출결 관리 방식들은 변화하는 교육 환경을 완전히 따라가지 못하고 여러 한계점을 드러내고 있다. 출석부 호출과 같은 전통적인 출석 방식은 많은 인원이 참여하는 수업에서 교사가 일일이 확인해야 하므로 시간 소모가 크고, 원격 수업에서는 적용이 어렵다. 비콘(Beacon)을 활용한 자동 출석 시스템도 별도의 하드웨어와 애플리케이션이 필요하며 거리나 기기 오작동 문제로 신뢰성이 떨어진다는[1][2]. 비교적 도입이 쉬운 QR/스마트코드 방식은 학생들이 QR코드나 출석 코드를 스캔하거나 입력해 출석을 인증하는 형태인데, 코드 공유를 통해 대리 출석이 이뤄질 우려가 있고 결국 교사가 이를 감독해야 하는 부담이 남는다[3].

이런 배경에서 출결 업무 자동화에 대한 요구가 높아지고 있다. 한 설문에서 교사의 75%가 출결 관리를 인공지능을 통한 업무 경감 분야로 꼽았으며, 이는 반복적이고 시간 소모적인 출결 업무를 자동화해 교사가 수업과 학생 지도에 집중할 수 있도록 지원하는 기술의 필요성을 보여준다[4].

이러한 요구에 부응하기 위해 본 논문은 Zulip 기반 챗봇 출결 관리 시스템을 제안한다. Zulip은 오픈소스 팀 채팅 플랫폼으로, 이를 수업용 보조 챗봇 시스템으로 활용하였다. 제안하는 챗봇은 수업 시간 동안 학생들에게 랜덤한 시점에 DM(다이렉트 메시지)을 보내 출석 확인을 시도한다. 주어진 시간 내 답변시 출석을 DB에 기록한다. 이러한 방식은 학생들이 언제 출석 확인 메시지를 받을지 모르기 때문에 수업 내내 긴장감을 갖고 집중하도록 유도하는 효과가 있다. 동시에 교사가 개입하지 않아도 실시간으로 출

석 여부가 파악되므로, 출결 확인 과정이 자동화되어 교사의 부담을 크게 덜어준다.

본 시스템은 LangChain 기반의 LangGraph 프레임워크를 활용해 구현되었다[7]. LangGraph는 대화 흐름을 조건에 따라 분기하거나 반복할 수 있는 기능을 제공하며, 도구(tool) 호출을 통해 외부 출석 데이터베이스와 연동되는 구조를 갖는다. 이러한 구조는 복잡한 교육 행정 절차의 자동화를 가능하다. 이에 본 연구는 LangGraph 기반의 대화형 에이전트를 활용하여 출결 관리 자동화를 구현하고, 이를 통해 교사의 업무 부담을 경감하는 동시에 출결 관리의 효율성과 신뢰성을 제고하는 것을 목표로 한다.

### 2. 시스템 아키텍처 및 DB

#### 2.1 시스템 아키텍처

[그림 1]은 챗봇 시스템의 전반적인 아키텍처를 보여준다. 사용자는 메시지를 보내고, 해당 메시지는 Zulip API를 통해 수신된다. Zulip API는 메시지를 Docker Compose로 구성된 AppServer에 request하고, AppServer는 이에 대한 답변을 생성한 후 다시 response한다.



Zulip에서 사용자가 전송한 메시지는 JSON 형식으로 구성되며, 주로 메시지의 고유 ID(id), 발신자 이름과 이메일(sender\_full\_name, sender\_email), 메시지 내용(content), 수신자 정보(recipient\_id), 전송 시각(timestamp), 메시지 유형(type) 등 핵심 정보를 포함한다.


사용자가 메시지를 보내면, Request Verifier가 해당 메시지가 관리자 명령어인지 여부와 발신자가 관리자 이메일인지 여부를 판별한다. 관리자 명령어는 query injection attack을 방지하기 위해 정확히 일치하는 명령어만 처리하도록 설계되어 있다. 주요 관리자 명령어로는 오늘 출석자 조회, 오늘 결석자 조회, 출석시간설정, 출석예약확인, 출석

The diagram illustrates the system architecture. At the top, a group of users interacts with a **Zulip** chat interface via email icons. A red arrow indicates a request from Zulip to the **Request Verifier** module. The **Request Verifier** module is part of the **App Server** and branches into two paths: one leading to the **LangGraph** workflow and another to a sequence of modules: **Administrator command**, **Attendance Scheduler**, and **Monitoring Module**. The **LangGraph** workflow includes an **Agent** (represented by a robot icon) which contains a **Chatbot**, **tools**, and an **\_end\_** node. A **logging** module is connected to the **tools** node and a **DB** (database) icon. A green arrow shows a response path from the **Monitoring Module** back to the **App Server**. A legend indicates that red arrows represent requests and green arrows represent responses.

이 중 '출석시간설정' 기능은 학생들의 랜덤한 출석 시간을 1, 2차로 구분하여 9:00~10:00, 10:30~11:30의 1시간 구간을 학생 수만큼 균등하게 나눈 후 학생마다 무작위 시각을 하나씩 매칭하는 방식으로 출석 시간을 설정하며, 해당 시각은 전역변수에 저장된다. 이후 Monitoring Module이 활성화되어 10초 간격으로 현재 시각을 감시하며, 전역 변수에 저장된 출석 요청 시간이 현재 시각을 지난 경우 해당 학생에게 즉시 Zulip DM을 전송한다. 마지막 출석 요청 시간이 지나면 Monitoring Module은 자동으로 종료되어 리소스 낭비를 방지한다.

## 2.2 데이터 저장소 및 테이블 설계

students		attendance	
email 	varchar	id 	integer
name	varchar	email	varchar NN
student_id	varchar	timestamp	datetime



[그림 2]와 같이 데이터베이스는 총 두 개의 핵심 테이블로 구성된다. students 테이블은 학생의 고유 정보를 저장하며, student\_id를 기본 키로 사용한다. attendance 테이블은 학생의 출석 응답을 기록하며, id는 각 출석 응답의 고유 식별자이다. 이 테이블의 email 필드는 students 테이블의 email을 참조한다.

### 3.1 실험 개요

### 3.2 실험 결과

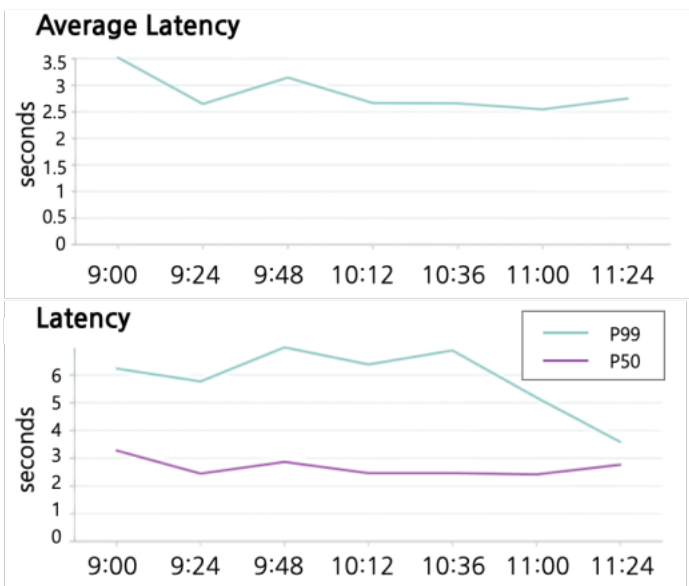
출석 응답 분석 결과, '출석'(639건) 외에도 '출책'(33건), '츠스', '츄르식', '출책해주세요' 등 총 30여 종의 비정형 표현이 관찰되었다. 시스템은 변형된 표현을 정확히 출석이라고 인식하고 출석 체크 도구를 호출했다. "fucking 출석 체크"와 같은 일부 비속어나 '출책할래용', '출석할꾸양', '출책에엥에엥에머리나노어리만어리나노오린다노야'와 같은 장난스러운 메시지에서 출석 의사가 분명한 경우 정확히 출석 체크 도구를 호출했다. 전체 응답에 대한 출석 체크 도구 호출의 정확도는 100%를 기록했으며, 이는 OpenAI API 기반 LLM이 키워드 매칭이 아닌 자연어 이해 기반으로 작동함을 보여주는 결과다.



[그림 3] 출석 메시지별 시도 및 성공 횟수(상위 10개)

출석 기록의 정확도는 100%였으나, 출석 가능 시간대를 분 단위로 안내한 반면 서버는 초 단위로 처리하여 몇 건의 시간 초과 사례가 발생하였다. 응답 지연 시간은 평균적으로 안정적인 수준이었으며, LLM 응답의 최대 지연 시간은 5.33초, 전체 프로세스의 최대 지연은 7.1초로 측정되었다. 이는 시스템이 실시간 수업 운영에 적용 가능할 정도로 안정적으로 작동함을 시사한다.

실험 중 특이 사례로는 한 사용자가 시스템을 반복 호출하여 784건의 메시지를 단독으로 전송한 경우가 있었으며, Zulip 이메일과 DB 상 이메일 불일치로 인해 출석 인식에 실패한 사례도 확인되었다. 이처럼 시스템은 다양한 표현을 유연하게 처리했으나, 시간 처리 방식과 사용자 메타데이터 관리 등 개선이 필요한 부분도 도출되었다. 본 실험은 LLM 기반 자연어 출석 처리 시스템이 실제 교육 환경에 효과적으로 적용될 수 있음을 실증적으로 보여준다.



[그림 4] Average, P50, P99 Latency

#### 4. 결론 및 향후 연구

본 연구를 통해 Zulip 기반 LangGraph 출석 자동화 시

스템은 실제 수업 환경에서 높은 신뢰성과 실시간 응답 성능을 갖춘 출석 처리 도구로 활용 가능성이 입증되었다. 평균 3초 내외의 응답 지연 시간은 실시간 수업 흐름에 영향을 미치지 않는 수준으로 평가되었다. 이는 빠른 피드백이 사용자 만족도 및 학습 몰입도를 향상시킨다는 기존 사용자 경험 연구와도 일치하는 결과이다[5][6]. 또한 본 시스템은 키워드 매칭이 아닌 자연어 이해를 기반으로 동작하여, '출석', '출석이름', 오타나 감정 표현이 포함된 메시지 등 비정형 응답에 대해서도 정확히 출석 체크 도구를 호출하며, 100%의 출석 처리를 정확도를 보였다. LangGraph 기반 접근법이 실제 언어 사용 환경에서 보다 견고한 인식 성능을 발휘할 수 있음을 보여준다.

한편, 실험을 통해 개선이 필요한 여러 지점도 확인되었다. Zulip 사용자 이메일 변경이 DB에 실시간 반영되지 않아 출석 처리가 누락된 사례, 시스템 혼란을 유발할 수 있는 대량 반복 메시지 전송, 그리고 욕설이 포함된 메시지 처리 문제는 사용자 행위 기반 예외 상황에 대한 별도 대응이 필요함을 시사한다. 이를 해결하기 위해서는 메시지 빈도 제한(rate limiting)을 적용하여 서버 부하를 방지하고, Azure OpenAI의 콘텐츠 필터링과 유사한 부적절 언어 탐지 기능을 도입해 대화 품질을 확보해야 할 것이다[8]. 또한 시분초 단위의 출석 가능 시간을 명확히 안내하고, Zulip 계정 정보와 학생 DB 간 동기화 기능을 구현함으로써 사용자 혼란을 줄일 수 있을 것으로 기대된다.

향후 연구에서는 LangGraph의 메모리 기능을 기반으로 사용자의 이전 상호작용 맥락을 반영한 질의응답 기능을 구현할 계획이다. 예를 들어 "오늘 출석했나요?"와 같은 질문에 과거 대화를 바탕으로 응답할 수 있는 기능은 챗봇의 일관성과 신뢰성을 향상시킬 수 있다. 또한 시험, 실습 성적 및 출석 데이터를 통합하여, 학생이 자신의 학습 현황을 실시간으로 조회하고 피드백을 받을 수 있는 자기주도 학습 지원 에이전트로 확장하는 방안도 고려 중이다[7]. 실제 사례에서도 출석 및 성적 정보를 제공하는 챗봇이 학업 진단 및 동기 부여에 긍정적인 영향을 미친 바 있으며, 본 시스템 역시 학습 맞춤화, 빠른 피드백, 학습 관리 자동화 측면에서 교육적 활용 가치가 높다고 판단된다[5][6].

#### 5. 참고문헌

- [1] 안성우, "BLE 기반의 비콘을 이용한 스마트 출석 확인 시스템," 『한국전자통신학회 논문지』, 제11권 제2호, pp. 209-214, 2016.
- [2] 이재홍, 채건희, 임근영, 설지환, 최성민, 임성욱, "비콘과 생체인식을 결합한 출석인증 시스템," 『한국차세대컴퓨팅학회 논문지』, 제14권 제2호, pp. 24-32, 2018.
- [3] 박선주, "QR코드를 활용한 스마트폰 기반 출석체크 시스템," 『정보교육학회논문지』, 제18권 제2호, pp. 325-334, 2014.
- [4] 김한나, 강신천, "인공지능을 활용한 교사 업무 경감 방안 연구," 『한국컴퓨터교육학회 학술발표논문집』, 제27권 제2호, pp. 165-171, 2023.

- [5] Chaiprasurt, C., Amornchewin, R., & Kunpitak, P. (2022). *Using motivation to improve learning achievement with a chatbot in blended learning*. World Journal on Educational Technology: Current Issues, 14(4), 1133–1151. DOI: 10.18844/wjet.v14i4.6592.
- [6] Labadze, L., Grigolia, M., & Machaidze, L. (2023). *Role of AI chatbots in education: systematic literature review*. International Journal of Educational Technology in Higher Education, 20, Article 56. DOI: 10.1186/s41239-023-00426-1.
- [7] LangChain, "LangGraph: Multi-Agent Workflows," *LangChain Official Blog*, Jan. 23, 2024. [Online]. Available: <https://blog.langchain.dev/langgraph-multi-agent-workflows/>
- [8] Microsoft, "Azure OpenAI Service content filtering," *Microsoft Learn*. [Online]. Available: <https://learn.microsoft.com/en-us/azure/ai-services/openai/concepts/content-filter>. [Accessed: May 6, 2025].