

인공지능 모델 통합 제어를 위한 LLM 기반 지능형 스마트팜 에이전트 시스템 구현 (LLM-based Agent System for Integrated AI Model Control in Smart Farms)

강태욱*, 신창선**, 박장우**, 박철영**

(Tae Wook Kang, Chang Sun Shin, Jang Woo Park, ChulYoung Park)

요약

스마트팜은 센서 모니터링을 넘어 인공지능(AI)과 대형 언어 모델(LLM)을 활용한 자율 제어로 진화하고 있다. 그러나 기존 시스템은 임계값 기반의 정적 제어에 의존하여, 실시간 환경 변화와 작물 생육 상태에 유연하게 대응하지 못하는 한계가 있다. 본 연구는 자연어 명령과 질의를 해석해 관련 제어 함수를 동적으로 선택·실행하고, 설명 가능한 자연어 응답을 생성하는 LLM 기반 함수 호출(Function Calling) 에이전트를 제안한다. LangChain과 Ollama를 이용해 센서 정보 조회, 생육 예측, 장비 제어 등의 기능을 통합했으며, RAG(Retrieval-Augmented Generation)의 환각(hallucination)을 줄이고자 도구 중심의 Agentic RAG 구조를 적용했다. 제안 시스템의 성능을 자체 데이터셋으로 평가한 결과 Macro F1-score 0.82를 달성하여, 질의 의도를 함수로 변환하는 능력의 실효성을 실험적으로 검증했다. 본 연구는 음성 인식 등 멀티모달 인터페이스를 통해 자율 제어의 실용성을 확인했으며, 향후 작물 맞춤형 제어 및 보안 강화로의 발전 가능성을 제시한다.

■ 중심어 : 스마트팜 ; 대형 언어 모델(LLM) ; 자율 제어 ; 함수 호출 ; 멀티 모달 에이전트

Abstract

Smart farms are evolving beyond sensor monitoring toward autonomous control using artificial intelligence (AI) and large language models (LLMs). However, conventional systems relying on threshold-based static control lack the flexibility to adapt to real-time environmental changes and crop growth stages. This study proposes an LLM-based agent with Function Calling that interprets natural language commands to dynamically execute control functions and generate explainable responses. Using LangChain and Ollama, it integrates functions like sensor retrieval, growth prediction, and equipment control. To mitigate the hallucination issues of Retrieval-Augmented Generation (RAG), a tool-centric Agentic RAG architecture was applied. The system's performance was experimentally validated on a custom dataset, achieving a Macro F1-score of 0.82 and confirming its effectiveness in translating user intent into appropriate functions. The practicality of this approach was further demonstrated through a multimodal interface, including voice recognition. This research suggests future potential for crop-specific control and enhanced security.

■ keywords : Smart Farm ; Large Language Model (LLM) ; Autonomous Control ; Function Calling ; Multimodal Agent

I. 서 론

최근 스마트 농업은 단순 모니터링 중심의 1세대 시스템을 넘어, 센서 기반 자동제어가 가능한 2세대 스마트팜으로 발전했다. 그러나 이 단계의

자동화는 주로 사전에 설정된 임계값 기반의 정적인 제어에 머물러 있으며, 환경 변화나 작물의 생육 상태에 대한 실시간 해석과 동적 제어에는 한계가 존재한다[1].

이에 따라 AI, 빅데이터, 대형 언어 모델(LLM:

* 준회원, 순천대학교 전보통신공학전공 석사과정

** 정회원, 순천대학교 인공지능공학부 교수

이 논문은 2025년 국립순천대학교 학술연구비(과제번호: 2025-0301) 공모과제로 연구되었음.

접수일자 : 2025년 09월 15일

수정일자 : 1차 2025년 10월 13일, 2차 2025년 10월 28일

게재 확정일 : 2025년 11월 12일

교신저자 : 박철영 e-mail : cypark@scnu.ac.kr

Large Language Model)을 접목하여 의사결정 까지 자동화되는 3세대 스마트팜으로의 전환이 요구되고 있다. 이 새로운 패러다임에서는, 생육 예측 및 질의응답을 통해 작물 상태를 실시간으로 해석하고, 사용자의 단순 질의만으로도 자율형 제어 전략이 수립될 수 있다. 특히, 경험 중심의 의사결정을 넘어 데이터 기반의 예측형 관리를 통해 생산성과 품질을 안정적으로 유지하려는 연구가 진행되고 있다[2].

대형 언어 모델(LLM)은 자연어 이해 및 함수 호출(Function Calling) 기능을 통해, 사용자의 질문이나 요청을 해석하고 복잡한 작업을 자동화할 수 있는 새로운 방식의 인터페이스를 제공한다 [3-4]. 스마트팜에 LLM 기반 에이전트(Agent)를 도입하면, 사용자는 복잡한 설정 없이도 단순한 질의만으로 인공지능 모델을 통해 생육 상태를 점검하거나 장비를 제어할 수 있으며, 작물별 동적 제어 전략을 생성하는 것이 가능하다[5].

제안하는 시스템은 연구자 중심으로 개발된 인공지능 모델을 일반 농민이 직접 사용할 수 있도록 하는 LLM 함수 호출 기반의 사용자 중심 모델 배포 전략을 채택하였다. 이는 복잡한 모델 학습·검증 과정을 추상화하고, 배포 후에도 지속적으로 모니터링, 재학습, 성능 관리가 가능하도록 하는 MLOps 개념을 내포하고 있다. 특히 LLM 에이전트 구조는 사용자가 음성이나 텍스트로 작물의 상태를 묻는 것만으로도, 시스템이 적절한 함수를 선택·실행하고 분석 결과에 기반한 제어 명령을 생성하는 완전한 자동화 흐름을 제공한다[6].

본 논문에서는 이러한 LLM 기반 스마트팜 제어 시스템의 에이전트 구조, 함수 및 체인 구성, 전체 아키텍처를 제안하고, 이를 실제 환경에서 구현·평가한 결과를 제시한다. 특히, 기존의 전문가 중심 제어 환경에서 벗어나, 일반 농민이 직접 활용 가능한 형태로 인공지능 시스템을 경량화·자동화하는 설계 전략을 제시하고자 한다.

II. 관련 연구

1. 스마트팜

스마트팜 기술은 센서 기반 모니터링의 1세대에서 자동제어가 추가된 2세대를 거쳐, 최근 AI와 대규모 데이터 활용의 3세대로 진화하고 있다. 표 1은 각 세대의 핵심 특징과 한계를 요약한 것이다.

표 1. 스마트팜 세대별 비교

Table 1. Smart Farm Comparison by Generation

구분	1세대 스마트팜	2세대 스마트팜	3세대 스마트팜
핵심 기술	IoT 센서 기반 모니터링	IoT + 자동제어 시스템	AI + 벡터데이터 + LLM 통합
제어 방식	사람이 데이터 분석 후 수동 제어	센서 기반 임계값 자동제어	실시간 AI 판단 기반 자율 제어
작동 원리	환경 데이터 수집 및 대시보드 제공	사전 설정된 조건에 따라 제어 명령 실행	AI 분석 → 생육 예측 및 최적 제어 명령 자동 생성
사용자 역할	농민이 직접 판단 조작	설정값 조정 및 자동 시스템 감시	자연어 질의, 시스템이 함수 선택·제어 자동 수행
운영 방식	수동 운영	반자동 운영	무인 자율형 운영
분석 주체	사용자	일부 로직 내장형 장비	LLM 및 AI 모델 통합 분석
주요 한계	실시간 대응 어려움, 의존도 높음	환경 변화 적응력 부족, 유연성 낮음	학습 기반 유연 대응, 질의 응답형 운영
작용 예시	스마트 관수 시스템, 모니터링 앱	온실 자동 창문·펜 제어 시스템	LLM 기반 생육 분석 + 자연어 질의 장비 제어

현재 상용 시스템 다수는 사용자가 최종 결정을 내리는 2.5세대 과도기에 머물러, 환경 변화에 대한 자율 적응성과 복합 조건 최적 제어에 한계가 보고된다. 반면 3세대는 실시간 센서 데이터와 질의응답 인터페이스를 결합하여 분석-판단-제어를 시스템이 수행하는 무인 자율형 운영을 지향한다. 본 연구는 이러한 흐름 속에서 LLM 에이전트를 데이터·제어 자원과 직접 연동해, 비전문가도 자연어로 상태 진단·정보 탐색·장치 제어를 수행하도록 한다.

2. Agentic RAG

기존 RAG 구조는 문서 검색과 생성 단계를 결합하여 LLM의 지식 한계를 보완하지만[7], 검색

된 문서의 정확도나 의미 정합성에 따라 환각(hallucination)이 발생할 수 있다. 이에 비해 Agentic RAG는 LLM이 목표 분석-도구 선택-다단계 탐색을 수행하며, 필요 시 검색·DB·IoT 함수를 호출해 사실 검증과 정보 보강을 반복한다. 아래 표 2는 두 접근의 구조적 차이를 요약한다.

표 2. 임베딩 벡터 기반 RAG와 Agentic RAG 비교
table 2. Comparison of Embedding Vector-Based RAG and Agentic RAG

항목	임베딩 기반 RAG	Agentic RAG (Function 중심)
핵심	유사 문서 검색 → 응답	정확한 문서 도달을 위한 능동적 정
목적	생성	보 탐색
기술	임베딩 → 벡터 검색 → 목적 분석 → 툴 호출(Function) →	
구조	LLM 응답	반복 탐색 및 최종 응답
도구		있음 (검색 API, DB 질의, 함수 실행 등)
활용	없음 (passive)	
특징	정적 단일 질의-응답 흐름 reasoning 포함	능동적 계획, 다단계 도구 실행 및
스마트팜 응용	생육 정보 문서 응답	목적에 맞는 생육 정보 + 센서 데이터 분석 흐름 구성

예를 들어, "온도 상승에 따른 조치"와 같은 복합적인 질의에 대해, 일반적인 RAG는 관련 매뉴얼 문서를 검색하는 데 그치지만, Agentic RAG 기반 에이전트는 다단계 추론을 수행한다. 우선 외부 도구(Function)를 호출하여 현재 온실의 정확한 온도 데이터를 확보하고, 이어서 데이터베이스를 조회하여 해당 작물의 생육 단계에 맞는 최적 온도 범위를 확인한 뒤, 두 정보를 종합하여 환풍기 작동과 같은 구체적인 제어 명령을 도출하는 식이다. 이러한 접근법은 Yao 등이 제안한 ReAct(Reason + Act) 프레임워크와 맥을 같이 한다. ReAct는 LLM이 추론(Reasoning)과 행동(Action)을 상호 보완적으로 수행하도록 설계하여, 단순 연쇄적 사고(Chain-of-Thought) 방식만 사용할 때보다 복잡한 문제 해결 능력과 사실 기반 응답률을 크게 향상시키고 환각을 줄일 수 있음을 실증적으로 보여주었다.[8]

III LLM 기반 지능형 스마트팜 에이전트 시스템

1. 함수호출 기반 LLM 에이전트 구조 및 처리 흐름

제안된 LLM 에이전트는 사용자의 자연어 입력을 분석해 필요한 함수를 호출하고, 결과를 기반으로 응답을 생성한다. LangChain 프레임워크를 활용해 추론과 제어를 연계하며, 생육 분석·장비 제어·정보 조회를 수행한다.

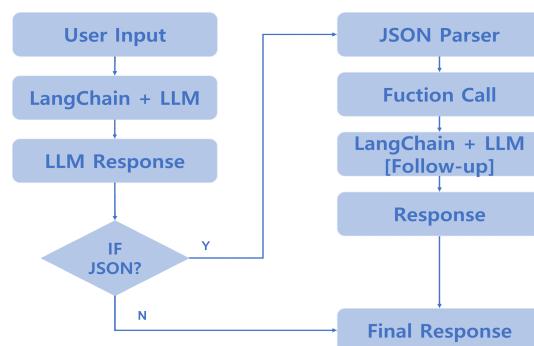


그림 1. Langchain 기반 LLM 에이전트 흐름도
Fig. 1. Langchain-Based LLM Agent Flowchart

그림 1은 사용자의 자연어 질의가 함수 호출(JSON) 형태로 변환되어 실행되는 전체 처리 과정을 시작화한 것이다. 사용자의 입력은 LangChain 기반 대형 언어 모델(Large Language Model, LLM)에 전달되며, 모델은 이를 분석해 적절한 응답을 생성한다. 응답이 함수 호출(JSON) 형식일 경우, 파서를 통해 함수명과 인자를 추출하고 해당 함수를 실행한다. 예를 들어, 작물 상태를 확인하는 질의가 입력되면 get_crop_status() 함수가 호출되며, 그 결과가 다시 LLM에 전달되어 설명 가능한 자연어 응답으로 변환된다. 반면 함수 호출이 필요 없는 단순 질의는 최초 생성된 LLM 응답이 그대로 반환된다.

표 3. 시스템 함수 목록

Table 3. List of System Functions

함수명	설명	파라미터 (args)
get_crop_disease	현재 작물의 질병 정보를 확인	없음
get_crop_growth_est	주어진 주차를 기준으로 작물의 엽장 길이를 예측	week(int, 선택, 기본값: 10)
get_crop_report	스마트팜 보고서 및 영농일지를 생성	없음
get_crop_status	작물의 이름, 작기, 품종, 과종일, 생육 일수, 생육단계를 조회	없음
get_device_status	스마트팜 제어 장치들의 상태를 조회	없음
get_document_link	문서(보고서, 영농일지 등)의 제목과 링크를 반환	keyword(str, 선택)
get_external_sensor	외부 온도, 습도, 풍속, 일사량 등 외부 환경 정보를 조회	없음
get_internal_sensor	실시간 환경 센서 정보를 조회	없음
get_now_time	현재 날짜와 시간을 반환 (형식: YYYY-MM-DD HH:MM).	없음
get_weather	지정된 위치의 현재 날씨를 반환	location(str, 필수)
set_device_control	특정 장치를 제어	device(str), action(str)

표 3은 시스템에서 정의된 제어 함수 목록으로, LLM이 호출 가능한 도구 범위를 나타낸다. 이 목록은 시스템의 기능 경계이자 평가 기준으로 활용된다. 각 함수들은 LangChain 기반 에이전트 내에서 사용 가능한 도구로 등록되어 있으며, LLM은 질의에 포함된 의도를 분석하여 필요한 함수를 자동으로 호출한다. 각 함수의 파라미터는 문맥을 통해 추론되어 자동으로 생성된다.

LLM 에이전트는 단순한 제어 명령 생성에 그치지 않고, 내부적으로 다양한 인공지능 모델을 연계하여 작물 상태를 분석하거나 생육을 예측하는 고차원적 기능도 수행한다[9]. 특히 다음 두 함수는 각각의 AI 모델을 내장하여 작물 분석 기능을 지원한다.

1) get_crop_disease 함수는 질의에 따라 YOLO 기반 병해충 탐지 모델을 호출하여, 스마트팜에서 수집된 작물 이미지를 입력으로 병해충 여부를 판별하고 결과를 자연어로 반환한다.

2) get_crop_growth_est 함수는 선형 회귀 기반 생육 예측 모델을 활용하여 생육 주차(week)를 인자로 엽장(葉長)을 추정하며, 기본값(10주차) 기준 예측도 지원한다. 이는 생육 진단 및 자동 보고서 생성에 활용 가능한 기초 데이터를 제공한다.

이러한 AI 모델 기반 도구들은 함수 호출 구조 내에서 LLM의 판단에 따라 선택적으로 호출되며, 사용자는 복잡한 모델 구조나 입력값을 직접 다루지 않고도 자연어 질의만으로 고급 분석 결과를 활용할 수 있다. 이는 기존의 수동 모델 활용 방식과 비교해 정보 접근성과 실행 편의성을 크게 향상시킨다.

특히 set_device_control 함수는 실제 제어 명령을 수행하는 핵심 도구로, 사전에 정의된 장치와 허용된 동작만을 대상으로 작동하며, 허용되지 않는 요청은 시스템 차원에서 차단되도록 프롬프트를 구성하였다.

2. LLM 에이전트 기반 스마트팜 시스템 아키텍처

본 연구에서 제안하는 지능형 스마트팜 LLM 에이전트 시스템은 사용자 질의 처리부터 센서 정보 수집, AI 기반 판단, 제어 명령 실행까지의 전체 프로세스를 하나의 자동화된 에이전트 구조로 통합한 것이다. 시스템 구성은 다음과 같은 네 개의 계층으로 구분된다: 사용자 인터페이스 계층, 응답 처리 계층, LLM 에이전트 계층, 환경 제어 계층이다.

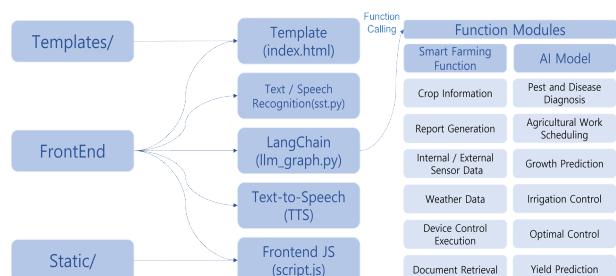


그림 2. 시스템 구성 다이어그램
Fig 2. System Configuration Diagram

그림 2는 이러한 네 계층 간의 상호 작용 관계와 데이터 흐름을 시각화한 것으로, 사용자의 자연어 질의가 LLM을 거쳐 제어 명령으로 전달되는 통합 구조를 보여준다.

(1) 사용자 인터페이스 계층

사용자는 웹 기반의 인터페이스를 통해 시스템에 자연어로 질의를 입력할 수 있으며, 사용자 편의를 위해 텍스트 입력뿐 아니라 음성(STT) 입력도 지원된다. 사용자가 요청한 질의는 FastAPI 서버를 통해 LLM 에이전트로 전달된다. 음성 질의의 경우 Whisper 모델을 활용해 텍스트로 변환되며, 응답 결과는 음성과 텍스트로 모두 출력 가능하다.

(2) 응답 처리 계층

FastAPI는 프론트엔드와 LLM 백엔드 사이의 중계 역할을 수행하며, WebSocket 기반의 비동기 스트리밍 통신을 통해 실시간 응답을 제공한다. 사용자의 질의는 해당 계층에서 전처리된 후 LLM 에이전트로 전달되며, LLM의 응답은 중간 단계 없이 사용자에게 즉시 표시된다. JSON 형태의 함수 호출 응답이 포함된 경우, 이를 파싱(Parsing)한 뒤 지정된 함수를 호출하고 결과를 다시 LLM에 전달하여 후속 응답을 생성한다.

(3) LLM 에이전트 계층

LangChain 에이전트는 각 기능을 Tool 객체로 등록해, 질의에 맞는 함수를 자동 선택·실행한다. 예를 들어 생육 진단 요청 시 get_crop_status, get_crop_report 등 함수가 실행되어 결과가 자연어로 요약된다. 함수 실행은 JSON 응답으로 결정되며, 사용자는 실행 로직을 전혀 인식하지 않고도 직관적인 질의만으로 복합 기능을 활용할 수 있다. 또한, 일부 함수는 내부적으로 인공지능 모델을 포함하고 있다. 구성된 함수 중 get_crop_disease는 YOLO 기반의 병해충 탐지 AI 모델을, get_crop_growth_est는 선형 회귀 기반의 생육 예측 모델을 구성했다.

(4) 환경 제어 계층

제어 계층에서는 IoT 장비 상태 조회 및 제어 명령 실행이 이루어진다. 센서 정보는 API 또는 MQTT를 통해 수집되며, set_device_control 함수를 통해 제어 명령이 전송된다. 해당 함수는 장치명(device)과 동작명(action)을 JSON 형태로 받아, 제어 가능 여부를 확인한 뒤 명령을 실행한다. 모든 제어는 사전 정의된 안전 범위 내에서만 이루어지며, LLM이 이를 판단하여 부적절한 명령은 차단하도록 프롬프트를 설계했다.

IV. 평가

1. 구현 환경

제안된 스마트팜 LLM 에이전트 시스템은 고성능 로컬 서버 및 경량화된 컨테이너 환경에서 실행되도록 설계되었다. 전체 구성은 하드웨어, 소프트웨어, 모델, 실행 플랫폼의 네 가지 측면으로 구분된다.

표 5. 시스템 환경 구성

Table 5. System Environment Configuration

구분	항목	사양 / 내용
하드웨어	CPU	Intel Core i9-14900K
	GPU	NVIDIA RTX GPU (VRAM 24GB)
	메모리	64GB RAM
	운영체제	Ubuntu 22.04 LTS
소프트웨어	서버 프레임워크	FastAPI, WebSocket, Docker
	에이전트 프레임워크	LangChain + Ollama
	음성 처리	Whisper (STT), edge-tts (TTS)
	이미지 분석	Ollama Vision API (base64 이미지 입력)
사용 모델	언어 모델	Gemma3:27b(멀티모달 지원 LLM)
에이전트	배포 방식	Ollama 기반 로컬 실행 (Docker 컨테이너)
	기능 통합	Function Calling 기반 도구 자동 호출
구조	함수 구성	functions/디렉토리에서 동적 로딩 및 등록

본 연구에서는 Google DeepMind의 오픈소스 언어 모델 Gemma3(27B 파라미터)를 기반 모델로 채택하였다. Gemma3는 128,000톤큰 컨텍스트 윈도우와 함수 호출(Function Calling) 기능을 내장하여, 장기 시계열 데이터 처리와 도구 제어에 모두 적합하다. 단일 GPU(24GB VRAM)에서도 안정적으로 구동되어 현장 실증 환경에 활용 가능하다[11].

2. 평가 기준 및 방법

본 평가는 통계적 일반화를 목적으로 하지 않으며, LLM 기반 함수 호출 구조에서 플래너(Planner) 모듈이 실제 운영 질의에 대해 올바른 질의 - 도구 매핑을 수행하는지를 기능적 측면에서 검증하는 데 초점을 두었다.

데이터셋은 비록 규모가 제한적이지만, 실제 스마트팜 운영에서 빈번하게 발생하는 핵심 질의 유형을 균등하게 포함해 포괄성을 확보하였으며, 동시에 본 시스템이 처리 가능한 함수 호출 범위(operational boundary)를 충실히 반영하도록 구성되었다.

이를 통해 시스템이 실제 환경에서 요구되는 질의 - 도구 매핑을 안정적으로 수행할 수 있는지를 객관적으로 진단하였다.

평가에는 스마트팜 운영 시나리오에서 발생할 수 있는 32개의 한국어 질의 - 정답 세트를 사용하였다. 각 질의는 표 3의 11개 시스템 함수 중 하나로 매핑되며, “밖 온도랑 습도 알려줘”는 get_external_sensor, “현재 창문 개도랑 커튼 상태 알려줘”는 get_device_status로 대응된다. 데이터셋은 질의 - 함수 매핑 정확도 검증을 위해 구축되었으며, 네 가지 질의 유형(환경 조회, 장치 제어, 생육 분석, 병해충 예측)을 균등하게 포함한다.

예측 결과와 정답 함수를 비교해 Precision, Recall, F1-score, Argument Accuracy를 측정하였다. Micro F1은 전체 결과, Macro F1은 질의별 평균으로 산출하였다.

표 6. 함수 호출 성능 평가 결과

table 6. Function Calling Performance Evaluation Results

Characteristic	Value
Micro Precision	0.667
Micro Recall	0.914
Micro F1-score	0.771
Macro F1-score	0.819
Argument Accuracy (Overlap)	0.969
True Positives (TP)	32
False Positives (FP)	16
False Negatives (FN)	3

표 6은 각 평가 지표의 수치를 요약한 것으로, 모델의 함수 선택 및 인자 해석 능력을 정량적으로 보여준다. 실험 결과, Micro F1-score는 0.77, Macro F1-score는 0.82로 전반적으로 높은 함수 선택 정확도를 보였다. 특히 Argument Accuracy가 96.9%로 나타나, 함수 선택 이후 인자 파싱 단계의 안정성이 검증되었다. 한편 Recall이 Precision보다 높게 나타나, 모델이 사용자의 의도를 대부분 포착하였으나 일부 불필요한 함수 호출이 발생했음을 의미한다.

3. 사례 세부 분석

사례 분석은 표 6의 평가 결과를 기반으로 수행하였다. 전체 32개의 질의 - 정답 세트를 스마트팜 제어 기능의 성격에 따라 네 유형(환경 조회, 장치 제어, 생육 분석, 병해충 예측)으로 분류하였다. 각 질의별 예측과 정답을 비교해 일치·부분 일치·오류로 구분하고, 호출 로그와 추론 과정을 검토해 주요 오류 원인(비정의 호출, 개념 혼동, 과잉 호출)을 도출하였다.

가. 완전 일치 사례 (F1=1.0)

외부 환경 조회(get_external_sensor), 장치 상태 확인(get_device_status), 작물 생육 상태(get_crop_status), 병해 위험도(get_crop_disease) 등 단순 질의에서는 정답과 예측이 일치하였다. 이는 모델이 명시적 질의 - 도구 매핑에 강점을 가짐을 보여준다.

나. 오류 및 실패 사례

(1)비정의 함수 호출: 저온 피해 가능성 포함 질의에서 정답(get_crop_disease) 대신 정의되지 않은 get_weather를 생성·호출하여 도구 카탈로그 제약을 위반함

(2)메타 질문 처리 실패: “당신은 누구죠?”와 같은 메타성 질의에서 불필요한 함수 호출(get_now_time, get_crop_disease)이 추가되어

Precision 저하가 발생.

(3) 개념 혼동: 외부/내부 센서 구분 오류(예: CO₂ 와 습도 질의에서 external_sensor 대신 internal_sensor 호출).

(4) 과잉 호출 사례: “오늘 날짜와 작물 정보” 질의에서 정답은 get_crop_status 하나였으나, 예측 결과에 불필요한 get_now_time 함수가 포함됨.

다. 부분적 일치 사례

“스마트팜 상태 종합적으로 알려줘” 질의에서는 정답(4개 함수)과 동일한 호출을 예측했으나, 불필요하게 get_internal_sensor가 포함되어 Recall=1.0, Precision 저하로 이어졌다.

4. 논의

평가 결과, 플래너 모듈은 명시적 질의 상황에서 높은 신뢰도를 보였으며, 실제 스마트팜 운영 환경에서도 안정적인 질의-도구 매핑을 수행하였다. 특히 Argument Accuracy 96.9%는 환경 데이터 처리와 제어 명령 실행 측면에서 높은 실용성을 입증한다.

또한, 본 연구의 성능은 HammerBench의 82.9%~91.6% 범위와 유사하거나 일부 상회하는 수준으로[12], 제안한 LLM 기반 함수 호출 모듈이 국제적으로 검증된 벤치마크와 동등한 성능을 달성했음을 보여준다. 특히, 한국어 자연어 질의 - 함수 호출 변환을 자체 구축한 데이터셋에서 수행했다는 점에서, 국내 언어 환경에 최적화된 함수 호출 모델의 실현 가능성과 기술적 진보를 시사한다.

한편, 일부 질의에서 발생한 불필요한 함수 호출과 개념 혼동은 Precision 저하의 주요 원인으로 확인되었다. Micro Precision(0.667)은 과잉 탐지(over-detection)에 따른 결과이며, Recall(0.914)은 사용자의 의도를 대부분 포착했음을 의미한다. 이러한 현상은 기능 누락을 최소화하도록 설계된 보수적 구조의 영향으로 해석된다.

향후 연구에서는 이러한 한계를 보완하기 위해

프롬프트 엔지니어링 고도화, 후처리 규칙 정교화, 도메인 지식 기반 파인튜닝, 다중 함수 호출 최적화 등을 통해 Precision 향상과 모델의 안정성 강화를 추진할 예정이다.

V. 결 론

LLM의 함수호출(Function Calling) 기술을 통해 비전문가도 자연어만으로 스마트팜을 정밀 제어할 수 있는 자율형 에이전트의 구현 가능성을 실험적으로 입증하였다. 제안 시스템은 LangChain 기반으로 사용자 의도를 해석해 필요한 도구를 단계적으로 호출하고, 센서 데이터·생육 지식·병해충 예측·제어 명령을 통합한다. 또한 Gemma-3 모델과 STT/TTS 인터페이스를 결합해 멀티모달 및 음성 상호작용을 지원한다. 이러한 실행 지향 구조는 기존 문서 중심 RAG의 정적 검색 의존과 환각 문제를 완화하고, 현장 데이터에 근거한 응답 생성과 제어 판단을 가능하게 한다.

Planner 모듈을 32개 한국어 질의로 평가한 결과, Micro-F1 0.77, Macro-F1 0.82, Recall 0.91, Precision 0.67, Argument Accuracy 0.97을 기록하였다. 이는 시스템이 요구 기능을 안정적으로 포착하고, 선택된 함수의 인자 해석 또한 신뢰할 수 있음을 보여준다. 대표 시나리오(병해충·생육 분석, 온실 제어 지시)에 대한 정성 검토에서도 응답의 유용성과 현실 적용성이 양호하게 평가되어, 제안 접근이 비전문가 친화적 스마트팜 운영에 실질적으로 기여함을 확인하였다.

Precision 저하는 불필요한 함수 과잉 호출과 센서 구분 혼동에 기인한다. 평가가 32쌍의 자체 구축 데이터셋으로 수행되어 통계적 일반화에는 한계가 있으나, 실제 운영 질의 유형을 포괄하고 시스템의 함수 호출 범위를 반영한 초기 실증 검증용 데이터셋으로서 의의를 갖는다. 향후 연구에서는 다중 함수 선택의 최적화, 도메인 지식 기반 미세조정, 데이터 확장 및 실증 환경 검증을 통해 성능을 고도화할 계획이다.

본 연구의 결과는 축사 관리, 시설 재배, 스마트시티 인프라 등 다양한 IoT 응용으로 확장 가능하다. 특히 LLM 기반 함수 호출 구조는 복잡한 제어 로직을 자연어 수준으로 단순화해, 비전문가 접근성과 현장 활용성을 높이는 실질적 도구로 기능한다. 본 연구는 LLM을 매개로 데이터·지식·제어를 통합한 초기 실증 사례로서, 자율형 스마트팜으로의 전환을 가속할 수 있는 기술적 토대를 제시하였다.

REFERENCES

- [1] 농촌진흥청, “농사로 - 스마트팜 구성 단계”(2019). <https://www.nongsaro.go.kr/portal/ps/psv/psvr/psvre/curationDtl.ps?menuId=PS03352&srchCurationNo=1362> (accessed Aug., 6, 2025).
- [2] 이현병, 최인찬, 임동혁, 허정욱, 김태현, 이정민, 이정호, 박성진, 이태석, 백정현, “LLM과 RAG를 활용한 스마트팜 제어 및 관리 지원 시스템 설계 및 구현,” *한국콘텐츠학회논문지*, 제24권, 제12호, 65-76쪽, 2024년 12월
- [3] 정천수, “LLM 애플리케이션 아키텍처를 활용한 생성형 AI 서비스 구현: RAG모델과 LangChain 프레임워크 기반,” *지능정보연구*, 제29권, 제4호, 129-164쪽, 2023년 12월
- [4] 곽지호, 김민근, 윤채원, 정유철, “기술정책 요약보고서의 내용 깊이 강화를 위한 LLM 및 RAG 기반 보강 기법 성능 비교,” *Proceedings of KIIT Conference*, 323-326쪽, 2024년 5월
- [5] 김민준, 권준석, “LLM-MA(Large Language Model Multi Agent)를 활용한 소프트웨어 개발에 대한 연구 동향 분석,” *대한전자공학회 학술대회*, 2711-2714쪽, 2024년 6월
- [6] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin, W. X. Zhao, Z. Wei, and J. Wen, “A survey on large language model based autonomous agents,” *Front. Comput. Sci.*, vol. 18, no. 6, p. 186345, mar. 2024.
- [7] 구정은, 신승중, “개인화된 농업 AI 컴페니언 시스템을 위한 LLM 기반의 인터페이스 설계 및 구현 -RAG 모델과 LangChain 프레임워크 기반-,” *한국인터넷방송통신학회 논문지*, 제24권, 제6호, 85-91쪽, 2024년 12월
- [8] Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K. R., Cao, Y., “ReAct: Synergizing Reasoning and Acting in Language Models,” *International Conference on Learning Representations (ICLR)*, may. 2023.
- [9] 박철영, 강태욱, 신창선, “스마트 농업을 위한 YOLO 기반 작물 생육측정 모바일 애플리케이션 구현” 스마

트미디어저널, 제14권, 제4호, 32-28쪽, 2025년 4월
doi: <https://dx.doi.org/10.30693/SMJ.2025.14.4.32>

- [10] Google, “Gemma Function Calling – Developer Guide,” (2025). <https://ai.google.dev/gemma/docs/capabilities/function-calling> (accessed Aug., 6, 2025).
- [11] Google, “Gemma 3 model overview [Model card],” (2025). <https://ai.google.dev/gemma/docs/core?hl=ko> (accessed Aug., 6, 2025).
- [12] Wang J, Zhou J, Wang X, et al. HammerBench: Fine-Grained Function-Calling Evaluation in Real Mobile Assistant Scenarios. Findings of the Association for Computational Linguistics p. 3350 - 3376, Jul. 2025.

저자 소개

강태욱(준회원)



2025년 순천대학교 정보통신공학과 학사 졸업

2025년~현재 순천대학교 정보통신공학과 석사 재학

<주관심분야 : 딥러닝, 머신러닝, AIoT>

신창선(종신회원)



1996년 우석대학교 전산학과 학사 졸업.

1999년 한양대학교 컴퓨터 교육학과 석사 졸업.

2004년 원광대학교 컴퓨터공학과 박사 졸업.

2005년~현재 순천대학교 인공지능공

학부 교수

<주관심분야 : 머신러닝, 분산시스템>

박장우(정회원)



1987년 한양대학교 전자공학과 공학사

1989년 한양대학교 전자공학과 공학석사

1993년 한양대학교 전자공학과 공학박사

1995년~현재 순천대학교 인공지능공

학부 교수

<주관심분야 : 머신러닝, 딥러닝, 시계열 분석>

박철영(정회원)



2010년 순천대학교 정보통신공학과 학사 졸업.

2012년 순천대학교 정보통신공학과 석사 졸업.

2017년 순천대학교 정보통신공학과 박사 졸업.

2025년~현재 순천대학교 인공지능공

학부 조교수

<주관심분야 : 인공지능 응용시스템, 딥러닝, 머신러닝, 클라우드 컴퓨팅>