

자동 에이전트 워크플로우 생성에 대한 종합적 고찰:

최신 연구 동향과 한계점

김민지¹, 김나연¹, 이영한²

¹ 성신여자대학교 융합보안공학과 학부생

² 성신여자대학교 융합보안공학과 교수

{20200904, 20210915, yhlee}@sungshin.ac.kr

A Comprehensive Survey on Automatic Agentic Workflow Generation: Recent Advances and Limitations

Min-Ji Kim¹, Na-Yeon Kim¹, Younghan Lee¹

¹Dept. of Convergence Security Engineering, Sungshin Women's University

요 약

본 연구는 LLM 기반 에이전트 시스템의 자동화된 워크플로우 생성을 탐색하고 서로 다른 방식의 대표 프레임워크인 AFLOW 와 FLOW 를 비교 분석하였다. 실험을 통해 두 방식 모두 높은 자동화 성능을 달성하였지만 실행 환경에서의 통합성과 검증 절차 부족이라는 한계를 확인하였다. 이를 보완하기 위해 Java Spring Boot 기반의 실행 실험을 수행하였고 자동화된 구조 생성과 실행 검증이 가능함을 입증하였다. 본 연구는 향후 실행로그 기반 워크플로우 개선을 포함한 실용적 자동 워크플로우 설계 방향을 제시한다.

1. 서론

대규모 언어 모델(Large Language Model, LLM)의 발전은 복잡한 문제 해결 능력이 크게 향상되면서 이를 전문화된 에이전트들이 협업하는 에이전틱 워크플로우(agent workflow) 형태로 확장하려는 시도가 활발히 이루어지고 있다. [1] 그러나 이러한 워크플로우는 설계 및 조정 과정이 고도로 복잡하고 반복적이라는 한계가 지적되고 있다. [2] 이에 따라 최근 인간의 개입 없이도 작업 단계를 구성하고 조정하는 워크플로우 자동화에 대한 연구가 활발히 진행되고 있다.

본 연구는 상이한 전략을 기반으로 설계된 대표적 프레임워크인 AFLOW 와 FLOW 를 비교 분석한다. AFLOW 는 정적 구조 최적화를 FLOW 는 동적 실행 적응을 지향한다. 이에 따라 본 연구는 두 프레임워크의 구조적 차이와 실행 전략을 비교함으로써 현재 기술의 한계점을 진단하고 향후 통합적 발전 방향에 대한 시사점을 도출하고자 한다.

2. 자동 에이전트 워크플로우 생성

에이전틱 AI(Agentic AI)란 복잡한 목표를 달성하기 위해 독립적으로 의사결정을 내리고 동적으로 적응할 수 있는 인공지능 시스템으로 정의된다. [3] 에이전틱 워크플로우란 이러한 AI 에이전트들이 협업하여 복잡

한 과제를 해결하는 구조적 프레임워크를 의미한다. 일반적으로 전체 작업을 여러 하위 단계로 세분화하고 각 단계에 특화된 에이전트를 배치하여 실행하며 LLM 기반의 추론 능력을 활용해 토론과 상호작용을 통해 최종 의사결정을 도출한다. [1][3] 전통적인 에이전트 워크플로우는 프롬프트 설계나 역할 정의 과정에서 수작업이 다수 개입되기 때문에 다양한 분야로의 확장성과 실행 유연성 측면에서 한계가 존재한다. 이러한 한계를 극복하기 위해 등장한 자동 에이전트 워크플로우는 LLM 을 활용하여 작업 단계를 자동으로 구성하고 조정하며 자율적으로 운영되는 프로세스이다. [4]

3. AFLOW : Agentic Workflow 생성도구

3.1 구조/탐색원리

AFLOW 는 LLM 기반 에이전트 워크플로우를 자동 생성하는 탐색 기반 프레임워크로 전체 구조는 노드, 엣지, 오퍼레이터 세 구성 요소로 정의된다. 노드는 LLM 호출 또는 외부 도구 실행과 같은 작업 단위를 의미하며 엣지는 작업 간 실행 순서와 의존성을 코드화한 흐름, 오퍼레이터는 반복되는 대화 패턴(예: Ensemble, Review 등)을 모듈화한 구성이다. 이 구조를 기반으로 AFLOW 는 Monte

Carlo Tree Search (MCTS) 알고리즘을 활용하여 최적의 워크플로우를 탐색한다. 초기 워크플로우(W_0)에서 시작해 LLM 이 후보 노드를 확장하고 생성된 워크플로우를 실행한 뒤 성능을 평가하며 이 결과는 역전파(Backpropagation)를 통해 탐색 경로에 반영된다. Soft Mixed Probability Selection 기법을 적용함으로써 성능이 우수한 경로는 높은 확률로 그렇지 않은 경로도 일정 비율 포함시켜 탐색의 다양성과 안정성을 동시에 확보한다.

3.2 의의

AFLOW 는 LLM 기반 구조 생성 과정에서 탐색 효율성과 워크플로우 최적화 성능을 동시에 확보하였으며 비교적 작은 LLM 으로도 높은 정확도의 문제 해결이 가능함을 입증했다.

4. FLOW : 동적 워크플로우 대응 자동 생성

4.1 구조/탐색원리

FLOW 는 각 작업(Task)을 Activity-on-Vertex(AOV) 그래프로 구성하여 방향성 비순환 그래프(DAG) 형태로 정의한다. LLM 은 이 구조를 기반으로 병렬성과 의존성 복잡도를 고려하여 최적의 실행 흐름을 구성한다. 실행 중에는 각 작업의 상태 및 로그 정보를 수집하여 필요 시 구조를 부분적으로 수정하거나 새로운 작업을 삽입하는 방식으로 워크플로우를 동적으로 유지한다.

4.2 의의

FLOW 는 실행 중 피드백을 수용하여 워크플로우를 유연하게 조정할 수 있는 구조를 갖추었으며 작업 실패 상황에서도 전체 구조를 유지한 채 필요한 부분만을 수정할 수 있는 구조적 복원력(resilience)을 갖추고 있다.

5. 한계점 및 향후 연구방향 제시

AFLOW 와 FLOW 와 같은 접근은 정적 코드 블록 생성을 중심으로 설계되어 있어 실제 실행 환경과의 통합 측면에서는 한계가 존재한다. 현실의 개발 과제에서는 문법적 완전성뿐만 아니라 시스템 연동 및 사용자 상호작용까지 아우르는 품질 보장 절차가 필수적이다. 특히 웹서버 부트 실행, API 테스트, 의존성 관리 등은 코드 생성 이후 반드시 수반되어야 하는 핵심 단계들이지만, 현재 프레임워크는 이를 충분히 통합하고 있지 않다.

이를 보완하기 위해 본 연구에서는 FLOW 프레임워크를 기반으로 AOV 구조를 설계하고 Spring Boot 프로젝트를 자동 생성한 뒤 실제 서버 실행 여부를 검증하는 실험을 수행하였다.

자동 생성된 코드는 Controller.java 와 의존성 파일을 포함하며 LLM 은 이를 기반으로 실행 가능한 애플리케이션 구조를 구성하였다. 이 과정은 LLM 을 활용한 코드 파싱, 구조 검증, bootRun 실행 및 로그 수집 등으로 구성되며 전체 워크플로우는 그림 1 에 제

시된 실행 구조를 따른다.

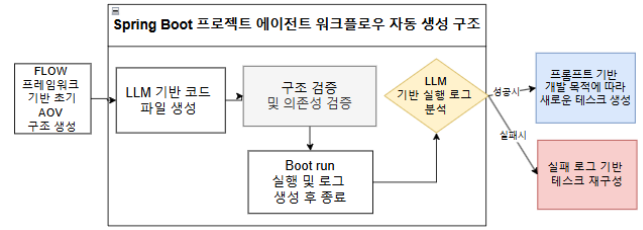


그림 1 : 스프링부트 설계 실험 자동 생성 구조

이 실험은 단순 코드 출력을 넘어 실행 가능한 실시간 워크플로우 자동 생성을 실증한 사례로 평가된다. 향후 해당 결과를 기반으로 테스트 자동화, API 검증, 예외 처리 등 후속 작업까지 통합함으로써 완전한 웹 개발 자동 워크플로우 설계로 확장할 수 있을 것으로 기대된다.

6. 결론

본 연구는 실제로 실행 가능한 워크플로우의 자동 생성 가능성을 FLOW 프레임워크 기반 실험을 통해 검증하였다. 특히 Spring Boot 환경에서 생성된 프로젝트가 실행 조건을 충족하고 실행 로그를 기반으로 후속 태스크를 유도할 수 있다는 점은 정적 생성 중심 프레임워크의 실용적 확장을 의미한다.

그러나 이 과정은 주로 단일 경로에서의 실행 기반 개선에 초점이 맞춰져 있어 다양한 실행 경로를 동적으로 탐색하는 전략까지는 포함되지 않았다. 이에 따라 향후 연구에서는 AFLOW 의 구조 최적화 탐색 기법과 FLOW 의 실행 적응 메커니즘을 통합한 하이브리드형 자동 워크플로우 설계가 요구된다.

참고문헌

- [1] Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, Bingnan Zheng, Bang Liu, Yuyu Luo, Chenglin Wu, "AFLOW: Automating Agentic Workflow Generation," Proceedings of the International Conference on Learning Representations (ICLR), Vienna, Austria, 2025, pp. 1–12.
- [2] Guangyao Chen, Siwei Dong, Yu Shu, Ge Zhang, Jaward Sesay, Börje F. Karlsson, Jie Fu, Yemin Shi, "AutoAgents: A Framework for Automatic Agent Generation," Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), Macao, 2024, pp. 25–32.
- [3] Panneer Selvam Viswanathan, "Agentic AI: A Comprehensive Framework for Autonomous Decision-Making Systems in Artificial Intelligence," International Journal of Computer Engineering and Technology, vol. 16, no. 1, pp. 862–880, 2025
- [4] Boye Niu, Yiliao Song, Kai Lian, Yifan Shen, Yu Yao, Kun Zhang, Tongliang Liu, "Flow: Modularized Agentic Workflow Automation," arXiv preprint, arXiv:2501.07834, 2025.

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 ICT 혁신인재 4.0 사업의 연구결과로 수행되었음 (IITP)