

Multi-hop 질의응답을 위한 자기 검증 구조 기반의 다중 에이전트 기법

Multi-Agent Scheme with Self-Verification for Multi-Hop Q&A

요약

대규모 언어 모델은 다양한 자연어 처리 작업에서 뛰어난 성능을 보이나, 여러 문서에 걸친 복잡한 추론이 필요한 멀티홉(Multi-Hop) 질의응답에서는 정보 검색 실패와 환각 현상의 한계를 드러낸다. 이를 해결하기 위해 ReAct 프레임워크가 제안되었으나, 단일 에이전트의 선형적 실행 구조는 초기 검색 오류가 수정 없이 후속 단계로 전파되어 최종 답변의 실패로 이어지는 구조적 취약점을 지니며, 자체 검증 메커니즘의 부재로 시스템 강건성이 저해된다. 본 연구는 이를 해결하기 위해 자기 검증(Self-Verification) 메커니즘을 갖춘 다중 에이전트 기법을 제안한다. 계획, 추론, 검색에 특화된 전문가 에이전트들이 협업하며, 수집된 정보의 신뢰성을 자기 검증 단계에서 평가한다. 검증 결과에 따라 추론을 진행하거나 탐색 전략을 수정하는 동적 교정 피드백 루프를 통해 오류를 사전에 차단하고 정확한 답변 생성 능력을 강화한다. HotpotQA 데이터셋 실험 결과, 제안 시스템은 기존 ReAct 대비 정확도(EM) 11.00점 향상, F1 점수 15.07점 향상을 달성했다. 본 연구는 전문화된 에이전트 간 유기적 협업과 구조화된 자기 검증이 멀티홉 추론의 정확도와 신뢰성 향상에 필수적임을 입증한다.

1. 서론

대규모 언어 모델은 방대한 사전 학습 지식을 바탕으로 자연어 이해 및 생성 분야에서 혁신적인 성과를 거두고 있다[1]. 특히, 모델이 학습하지 않은 최신 정보나 전문적인 지식을 활용하기 위해 외부 지식 베이스에서 관련 정보를 검색하여 입력으로 제공하는 검색 증강 생성(Retrieval-Augmented Generation, RAG) 기술이 주목받고 있다[2]. RAG는 LLM의 고질적인 문제인 환각(Hallucination) 현상을 완화하고 답변의 사실적 근거를 제시할 수 있어, 신뢰성 높은 질의응답 시스템 구축을 위한 핵심 기술로 자리 잡았다[3]. 답 시스템 구축을 위한 핵심 기술로 자리 잡았다[3].

그러나 단순한 사실 검색을 넘어, 여러 문서에 산재된 정보를 연결하여 복합적인 추론을 수행해야 하는 멀티홉(Multi-hop) 질의응답 태스크에서는 여전히 정확한 답변 생성에 한계가 존재한다[4]. 단일 질문에 대한 검색은 비교적 용이하지만, 추론 단계가 깊어질수록 검색된 정보의 정확도가 떨어지거나 문맥이 소실되는 문제가 빈번하게 발생하기 때문이다. 이러한 부정확한 중간 정보는 후속 추론 단계에 악영향을 미쳐 최종 답변의 품질을 심각하게 저하시키는 원인이 된다[5].

이러한 문제를 해결하기 위해 추론(Reasoning)과 행동(Acting)을 결합한 ReAct(Reasoning and Acting) 프레임워크가 제안되어 널리 활용되고 있다[6]. ReAct는 LLM이 스스로 사고 과정을 생성하고 필요한 도구를 호출하는 방식으로 외부 정보 활용 능력을 향상시켰다. 그러나 ReAct 역시 단일 에이전트의 선형적 실행 구조로 인해, 초기 검색 오류가 수정 없이 후속 단계로 전파되어 최종 답변의 실패로 이어지는 구조적 취약점을 가진다[7]. 또

한, 단일 모델에 계획, 추론, 검색 등 모든 부하가 집중되어 복잡한 문제일수록 인지 능력 저하 문제가 발생한다.

본 논문은 ReAct의 이러한 구조적 한계를 극복하기 위해, 자기 검증(Self-Verification) 및 동적 교정(Dynamic Correction)이 가능한 다중 에이전트 기법을 제안한다. 제안 기법은 LangGraph를 기반으로 계획(Planner), 추론(Reasoner), 검색(Searcher) 등으로 역할을 분산하여 전문화된 에이전트들이 협업하는 구조를 갖는다[8]. 특히, 에이전트 간의 상호작용을 순환형 그래프(Cyclic Graph)로 모델링함으로써, 수집된 정보가 불충분하다고 판단될 경우 즉시 피드백 루프를 작동시켜 탐색 전략을 수정(Refine Strategy)하거나 대안을 모색하는 자기 교정(Self-Correction) 메커니즘을 구현하였다. 이를 통해 기존 ReAct 방식의 고질적인 문제인 오류 전파를 효과적으로 차단하고, 초기 검색이 실패하더라도 반복적인 검증과 재시도를 통해 정확한 답변 생성 능력을 강화하였다.

본 연구의 주요 기여는 다음과 같다. 첫째, 전문화된 에이전트 간 유기적 협업 구조를 통해 단일 에이전트의 인지 과부하를 해결하였다. 둘째, 자기 검증 메커니즘을 통해 오류를 조기에 감지하고 차단함으로써 오류 전파 문제를 근본적으로 해결하였다. 셋째, 동적 교정 피드백 루프를 통해 초기 실패에서 복구하는 강건성을 확보하였다.

본 논문의 구성은 다음과 같다. 2장에서는 대규모 언어 모델을 활용한 에이전트 연구 동향과 멀티홉 질의응답 기술에 대한 관련 연구를 소개한다. 3장에서는 본 논문에서 제안하는 LangGraph 기반의 멀티 에이전트 기법의 아키텍처와, 에이전트 간의 유기적인 협업 및 자기 검증 알고리즘을 상세히 기술한다. 4장에서는 실험에 사용된 HotpotQA 데이터셋과 구체적인 실험 환경을 소개

하고, 기존 ReAct 모델과의 정량적 비교 실험을 통해 제안 기법의 성능 향상과 오류 복구 능력을 실증적으로 분석한다. 5장에서는 연구의 결론을 맺고 향후 연구 방향을 제시한다. 마지막 부록에서는 제안 모델과 베이스라인 모델의 실제 추론 경로(Reasoning Trajectory)를 시각적으로 비교한 정성적 분석 결과를 제시하여, 본문에서 제안 기법의 동작 원리를 구체적인 사례를 소개한다.

2. 관련 연구

본 장에서는 멀티홉 질의응답의 도전과제를 설명하고, 이를 해결하기 위한 기술 발전 과정을 설명한다. 특히 단일 에이전트 기반 ReAct 프레임 워크의 구조적 한계를 분석하고, 멀티 에이전트 협업 기술의 필요성을 설명한다.

2.1 멀티홉 질의응답

멀티홉 질의응답은 여러 문서에 퍼져있는 정보를 복합적으로 파악하여 추론해야 하는 복잡한 과제로, 크게 3가지 어려움을 고려해야 한다[4]. 첫째, 질문 분해(Question Decomposition)의 어려움으로 복잡한 질문을 적절한 하위 목표로 분해하고 실행 순서를 결정해야 한다[18]. 둘째, 순차적 정보 검색(Sequential Retrieval)의 어려움으로, 이전 검색 결과 기반의 후속 검색어 생성 과정에서 오류 누적 문제가 발생한다. 셋째, 방해 정보 선별(Distractor Filtering)의 어려움이다. HotpotQA는 정답과 무관한 방해 문단을 포함하여 노이즈 환경에서의 정확한 정보 선별을 요구한다[17].

기존 연구들은 주로 텍스트 매칭이나 단순 검색에 의존하여, 질문의 복잡한 의도 파악과 다단계 계획 수립에 한계를 보였다. 이를 극복하기 위해 대규모 언어 모델 기반 추론 기술과 에이전트 접근법이 등장했다.

2.2 대규모 언어 모델의 추론 능력

최근 GPT-4와 같은 대규모 언어 모델은 높은 수준의 자연어 이해 및 생성 능력을 보여준다[9]. 그러나 복잡한 산술 연산이나 다단계 논리 추론 문제에서 빈번한 오류를 범하며, 환각(Hallucination) 문제를 유발한다[10].

이를 해결하기 위해 중간 추론 단계를 생성하도록 유도하는 CoT(Chain-of-Thought) 프롬프팅 기법이 제안되었다[11]. CoT는 복잡한 문제를 중간 논리 단계로 분해하여 추론 성능을 향상시켰다. 그러나 모델 내부의 파라미터 지식에만 의존하므로, 최신 정보나 전문 지식이 요구되는 태스크에서는 한계를 보인다[12]. 이를 극복하기 위해 LLM이 외부 도구와 지식 베이스를 활용하는 에이전트 접근법이 등장했다.

2.3 단일 에이전트 추론: ReAct 프레임워크

LLM 기반 에이전트의 대표적 방법론은 ReAct 프레임 워크이다[6]. ReAct는 추론(Thought), 행동(Action), 관찰(Observation)을 반복하며 문제를 해결한다. 모델이 사고 과정을 생성하고, 필요시 외부 도구를 호출하며, 그 결과를 관찰하여 다음 단계를 결정한다.

ReAct는 외부 정보 검색을 통해 환각 현상을 줄이고 답

변의 사실적 근거를 제공할 수 있다. 그러나 단일 에이전트가 모든 과정을 수행하는 선형적 구조로 인해, 멀티홉 질의응답과 같은 복잡한 추론 태스크에서는 다음과 같은 한계를 보인다.

첫째, 오류 전파(Error Propagation) 문제이다. 초기 단계에서 잘못된 키워드로 검색하거나 부정확한 정보를 추출할 경우, 이를 수정할 기회 없이 이후의 모든 추론 과정이 오염되어 최종 답변 실패로 이어진다[7]. 둘째, 자기 교정(Self-Correction) 능력의 부재이다. 검색 결과가 불충분하거나 질문과 관련이 없더라도 이를 스스로 감지하고 전략을 정하는 피드백 루프가 미흡하다. 특히 방해 정보가 많은 멀티홉 질의응답 환경에서 잘못된 정보를 맹신하여 성능 저하로 이어진다[14]. 셋째, 정보 통합의 취약성이다. 검색된 여러 문서 간 관계 파악과 통합 능력이 제한적이어서 멀티홉 추론에서 성능 저하가 두드러진다.

2.3 멀티 에이전트

단일 에이전트의 인지 과부하(Cognitive Overload)와 구조적 경직성을 해결하기 위해, 최근에는 다수의 전문화된 에이전트가 협업하는 멀티 에이전트 시스템이 주목받고 있다[15]. 멀티 에이전트 시스템은 복잡한 태스크를 계획, 실행, 검증 등의 하위 역할로 분해하여 각 에이전트에게 할당함으로써 개별 모델의 성능을 최적화하고 인지 부하를 분산시킨다[16].

특히 LangGraph와 같은 그래프 기반의 에이전트 협업 도구의 등장은 에이전트 간의 상호작용을 순환형 그래프로 모델링하는 것을 가능하게 했다[8]. 이는 기존의 단방향 체인(구조와 달리, 특정 조건이 충족되지 않을 경우 이전 단계로 되돌아가거나 새로운 경로를 탐색하는 동적 제어 흐름을 구현할 수 있게 해준다. 본 연구에서는 이러한 멀티 에이전트 협업 구조를 활용하여, 검색 실패 시 계획을 수정하는 재계획 메커니즘을 구현한다.

2.4 멀티 에이전트 협업 시스템

ReAct의 한계점들을 해결하기 위해, 다수의 전문화된 에이전트가 협업하는 멀티 에이전트 시스템이 주목받고 있다[15]. 멀티 에이전트 접근법은 복잡한 태스크를 계획, 실행, 검증 등의 하위 역할로 분해하여 각 에이전트에 할당한다[16]. 이를 통해 첫째, 역할 전문화로 개별 성능을 최적화하고, 둘째, 에이전트 간 검증으로 오류를 조기 발견하며, 셋째, 인지 부하 분산으로 복잡한 문제에서도 안정적 성능을 유지한다. 특히 LangGraph와 같은 그래프 기반 오케스트레이션 도구는 에이전트 간 상호작용을 순환형 그래프(Cyclic Graph)로 모델링할 수 있게 했다[8]. 이는 기존 단방향 체인 구조와 달리, 특정 조건 미충족 시 이전 단계로 되돌아가거나 새로운 경로를 탐색하는 동적 제어 흐름을 구현한다. 본 연구는 멀티 에이전트 협업 구조를 활용하되, 자기 검증 메커니즘을 핵심 요소로 도입하여 오류 전파 문제를 근본적으로 해결하고, 검색 실패 시 계획을 수정하는 동적 교정 메커니즘을 구현한다.

3. 제안한 방법

본 장에서는 멀티홉 질의응답을 위한 자기 검증 기반 다중 에이전트 시스템의 구조와 동작 원리를 제시한다. 먼저 시스템 전체 구조를 소개하고, 각 에이전트의 역할과 협업 방식을 설명한다. 이어서 오류 전파를 차단하는 자기 검증 메커니즘과 검색 실패 시 전략을 수정하는 동적 교정 프로세스를 상세히 기술한다.

3.1 시스템 개요

제안 시스템은 전문화된 에이전트들이 협업하여 멀티홉 질문을 해결하는 구조를 갖는다. 시스템은 계획(Planner), 추론 및 제어(Reasoner), 답변 생성(Answer), 정보 검색(Searcher), 정보 추출(Extractor)의 5개 모듈로 구성된다. 각 에이전트는 공유 상태 메모리를 통해 현재 계획, 검색된 증거(Evidence), 수행 기록들을 실시간으로 공유한다. 기존 ReAct는 Thought, Action, Observation의 선형적 흐름을 가지나, 제안 시스템은 Reasoner가 수행 결과를 평가하고 필요시 탐색 전략을 수정하는 동적 교정 피드백 루프를 포함한다.

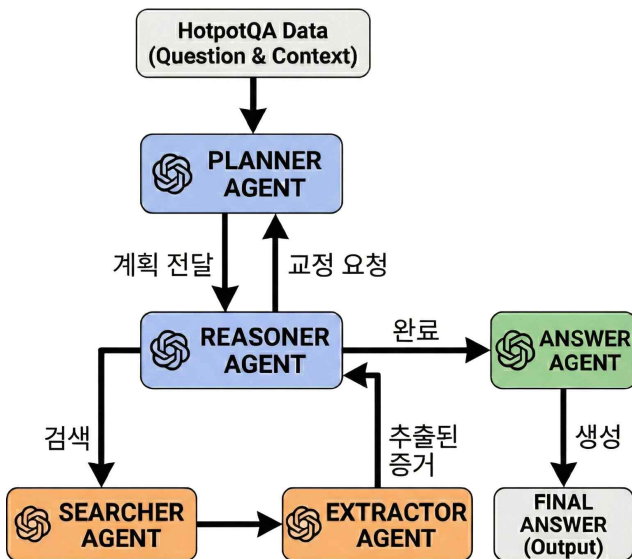


Figure 1. 제안 기법 개요도

3.2 에이전트 구성 및 역할

본 기법은 역할에 따라 인지 및 제어 계층(Cognitive & Control Layer)과 탐색 및 실행 계층(Retrieval & Execution Layer)으로 구분된다. 상위 계층인 Planner와 Reasoner는 문제 해결 전략을 수립하고 프로세스를 감독하며, 하위 계층인 Searcher와 Extractor는 실제 정보 탐색을 수행한다.

3.2.1 인지 및 제어 계층

이 계층은 시스템의 두뇌로서 문제 분해, 실행 제어, 결과 검증, 답변 생성을 담당한다. Planner Agent는 입력된 복잡한 멀티홉 질문을 해결 가능한 단위의 하위 목표로 분해하여 전체적인 로드맵을 설계한다. Reasoner와 동등한 수준의 인지적 협력 관계를 유지하며, 초기 계획

수립뿐만 아니라 실행 도중 Reasoner로부터 교정 요청(Correction Request)이 수신되면 실패 원인을 분석하여 탐색 경로를 재설정하거나 전략을 수정하는 등 유동적인 의사결정을 수행한다. Reasoner Agent는 시스템의 중앙 제어 장치로서 Planner의 계획을 관리하고 하위 에이전트 실행을 조율한다. 현재 단계 진행을 모니터링하며, 필요 시 Searcher와 Extractor를 호출하여 작업을 위임한다. 자기 검증 로직으로 수집 정보의 충분성을 평가하고, 불충분 시 재검색 지시나 Planner에 동적 교정을 요청하여 전략을 수정한다. Answer Agent는 추론의 최종 단계에서 호출되는 에이전트이다. Reasoner가 검증을 완료한 모든 증거와 단계별 추론 결과를 종합하여 사용자 질문에 대한 최종 답변을 생성한다. 단순히 정보를 나열하는 것이 아니라, 프롬프트에 정의된 규칙에 따라 질문의 유형을 분석하고 불필요한 서술을 배제한 명확한 최종 정답을 제공한다.

3.2.2 탐색 및 실행 계층

이 계층은 시스템의 손과 발로서 Reasoner의 지시에 따라 데이터셋 내에서 실제 정보를 찾고 추출한다.

Searcher Agent는 Reasoner의 지시에 따라 구체적인 문서를 찾아내는 특화된 탐색 에이전트이다. HotpotQA와 같이 다수의 방해 문단이 존재하는 환경을 고려하여, 주어진 문맥 중에서 현재 하위 목표와 가장 연관성이 높은 문서를 선별한다. 이때, 이전 단계의 추론 기록들을 문맥으로 활용하여 대명사가 지칭하는 대상을 추적함으로써 정확한 문서 선별을 지원한다. Extractor Agent는 Searcher가 선별한 문서 내에서 정답 도출에 필요한 핵심 증거만을 추출하는 실행 에이전트이다. 문서 전체 대신 질문과 직접 관련된 사실 문장을 정제하여 추출함으로써 노이즈를 제거하고 환각 가능성을 최소화한다. 추출 정보는 Reasoner에 반환되어 검증 및 최종 추론의 근거로 활용된다.

3.3 자기 검증 메커니즘

단일 에이전트 모델의 주요 실패 요인은 잘못 수집된 정보를 검증 없이 다음 단계 입력으로 사용하여 오류가 누적되는 것이다. 본 시스템은 오류 전파를 원천 차단하기 위해 Reasoner 내부에 필수적인 자기 검증 단계를 포함한다. 검증 프로세스는 Extractor의 정보 추출 직후 실행되며, 대규모 언어 모델을 활용하여 단순 키워드 매칭이 아닌 논리적 충분성을 기준으로 증거 유효성을 평가한다. 구체적으로 현재 단계의 하위 목표 질문과 추출된 증거를 대조하여 상호 연관성을 분석한다. 이후 해당 증거가 질문에 대한 직접적 해답이나 유의미한 추론 단서를 포함하는지 판단하는 충분성 평가를 수행한다. 문서 내 명확한 정보가 부재하거나 질문 의도와 무관할 경우, 시스템은 즉시 불충분하다고 판정하여 잘못된 정보가 다음 단계로 전이되는 것을 방지한다.

3.4 동적 교정 및 순환 구조

단순 재검색으로 해결되지 않는 구조적 정보 부재나 초기 계획 오류를 극복하기 위해, 본 시스템은 LangGraph의 순환형 구조를 활용한 동적 교정 메커니즘을 제안한다. Figure 2는 Planner와 Reasoner 간 교정 및 전략 수정 상호작용 루프를 나타낸다.



Figure 2. 계획-추론 에이전트 간 동적 수정 루프

3.4.1 교정 트리거

Reasoner는 추론 과정에서 현재 계획 수행이 불가능하다고 판단 시 Planner에 동적 교정을 요청한다. 교정 요청은 두 가지 상황에서 실행된다. 첫째, 특정 하위 목표 달성을 위해 임계값 이상의 재시도를 수행했으나 유효한 정보를 추출하지 못한 경우이다. 둘째, 현재 문맥 내 접근 가능한 모든 가용 문서를 탐색했으나 질문 해결에 필요한 결정적 증거를 확보하지 못한 경우, 시스템은 이를 구조적 정보 부재로 인식하고 즉시 전략 수정을 요구한다.

3.4.2 교정 및 전략 수정

교정 요청을 수신한 Planner는 현재까지의 수행 이력을 분석하여 실패 원인을 규명하고 탐색 전략을 수정한다. 단순 질문의 형식이 아닌, 다음의 고도화된 상태 제어 전략을 적용한다.

성공 상태의 보존 (State Preservation)는 기 수행된 하위 단계 중 성공적으로 검증된 결과값(Entity, Fact)은 동결하여 보존한다. 이는 재계획 과정에서 이미 획득한 정보의 손실을 방지하고, 불필요한 중복 연산을 최소화하여 효율성을 높이기 위함이다. 탐색 공간의 재설정 (Search Space Reset)는 실패가 발생한 단계에 한하여, 이전에 '관련 없음'으로 분류되어 배제되었던 문서들의 상태를 초기화한다. 이는 새로운 관점이나 키워드가 적용될 경우, 기각되었던 문서에서 유효한 증거를 재발견할 가능성을 열어두기 위함이다. 부분 정보를 활용한 추론 (Inference with Partial Information)는 비교 유형의 질문에서 일부 대상의 정보가 누락된 경우, 무한 루프를 방지하기 위해 확보된 단일 대상의 정보와 논리적 소거법을 결합하여 최선의 답을 유추하는 전략을 수행한다.

본 시스템의 동적 교정 메커니즘은 선형적 실행 흐름을 탈피한 피드백 루프를 형성하며, 정답 도출이나 최대 반복 횟수 도달까지 유기적으로 순환한다. 결과적으로 시스템은 고정된 계획에 얽매이지 않고 상황 변화에 유연하게 대응하는 적응형 추론 능력을 확보한다.

4. 실험 및 결과

본 장에서는 제안 시스템의 성능을 검증하기 위한 실험 환경과 데이터셋을 소개하고, 기존 ReAct 모델과의 비교 실험 결과를 제시한다. 정량적 성능 비교를 통해 제안 시스템의 우수성을 입증하고, 성능 향상의 주요 원인을 분석한다.

4.1 실험 환경 및 데이터셋

본 연구는 대표적인 멀티홉 질의응답 벤치마크인 HotpotQA의 Distractor Setting을 사용하였다. HotpotQA Dev 세트에서 무작위로 추출한 100개 샘플로 실험을 수행하였으며, 공정한 비교를 위해 비교 방법도 동일한 샘플과 10개 문단 환경에서 검색하도록 설정하였다. 모든 실험은 GPT-4o-mini를 베이스 모델로 사용하였다. 평가 지표는 정답과의 완전 일치 비율인 EM(Exact Match)과 정답 텍스트와의 중복 정도를 측정하는 F1 Score를 사용하였다.

4.2 비교 실험 결과

제안하는 멀티 에이전트 시스템과 기존 ReAct 모델의 성능 비교 결과는 Table 1과 같다.

모델(Model)	EM (Exact Match)	F1
Baseline (ReAct)	46.00	55.31
Ours (Multi-Agent)	57.00	70.38

Table 1. 정량 평가 비교 결과

제안 기법은 베이스라인 대비 EM에서 11.00, F1에서 15.07 향상을 달성하였다. 이는 제안 시스템이 기존 방식보다 훨씬 높은 정확도로 정답을 도출함을 입증한다.

이러한 성능 격차는 성능 향상이 다중 에이전트 구조와 자기 검증 메커니즘에 기인한다. 주요 원인은 다음과 같다. 기존 ReAct는 초기 검색 오류 발생 시 이를 수정할 기회 없이 전체 추론이 오염되는 오류 전파(에 취약하다. 반면 제안 시스템은 Reasoner가 수집 증거의 불충분함을 감지하면 즉시 Planner가 검색 전략을 수정하는 자기 검증 및 동적 교정 과정을 통해 오류 전파를 원천 차단한다. 또한 교정 단계에서 실패한 검색 이력을 초기화하고 새로운 관점에서 문서를 재검색하는 전략이 주요하였다. 이를 통해 초기에 간과된 문서에서 유효한 정보를 발견할 수 있었다. 추가적으로 비교 유형 질문에서 모든 정보를 완벽히 수집하지 하더라도, 확보된 정보와 논리적 소거법을 결합하여 정답을 유추하는 강건성을 보였다.

결과적으로 제안 시스템은 단일 에이전트가 처리하기 어려운 복잡한 다단계 추론 문제에서 구조적 상호작용과 검증 루프를 통해 월등한 신뢰성과 정답률을 확인할 수 있었다.

5. 결론

본 연구는 멀티홉 질의응답의 오류 전파 문제를 해결하기 위해 자기 검증 및 동적 교정 기반 다중 에이전트 기법을 제안하였다. 제안 기법은 전문화된 에이전트 간 협업과 순환형 그래프 구조를 통해 검색 실패 시 즉시 전략을 수정하는 메커니즘을 구현하였다. HotpotQA 실험 결과, ReAct 대비 EM 11.00점(23.9% 향상), F1 15.07점 향상을 달성하였다. 본 연구는 에이전트 간 구조화된 상호작용과 검증 절차가 멀티홉 추론의 정확도와 신뢰성 향상에 필수적임을 입증하였다. 향후 통신 비용 최적화, 개방형 도메인 확장, 멀티모달 처리로 연구를 확장할 예정이다.

[참고 문헌]

- [1] T. Brown et al., "Language Models are Few-Shot Learners," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 33, pp. 1877 - 1901, 2020.
- [2] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 33, pp. 9459 - 9474, 2020.
- [3] Z. Ji et al., "Survey of Hallucination in Natural Language Generation," ACM Computing Surveys, vol. 55, no. 12, pp. 1 - 38, 2023.
- [4] Z. Yang et al., "HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering," in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), pp. 2369 - 2380, 2018.
- [5] A. Mallen et al., "When Not to Trust Language Models: Investigating Effectiveness and Limitations of Parametric and Non-Parametric Memories," in Proc. Annu. Meeting Assoc. Comput. Linguistics (ACL), pp. 9802 - 9822, 2023.
- [6] S. Yao et al., "ReAct: Synergizing Reasoning and Acting in Language Models," in Proc. Int. Conf. Learn. Represent. (ICLR), 2023.
- [7] N. Shinn et al., "Reflexion: Language Agents with Verbal Reinforcement Learning," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 36, 2023.
- [8] Q. Wu et al., "AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation," arXiv preprint arXiv:2308.08155, 2023.
- [9] OpenAI, "GPT-4 Technical Report," arXiv preprint arXiv:2303.08774, 2023.
- [10] L. Huang et al., "A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions," arXiv preprint arXiv:2311.05232, 2023.
- [11] J. Wei et al., "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 35, pp. 24824 - 24837, 2022.
- [12] N. Kandpal et al., "Large Language Models Struggle to Learn Long-Tail Knowledge," in Proc. Int. Conf. Mach. Learn. (ICML), pp. 15696 - 15715, 2023.
- [13] T. Schick et al., "Toolformer: Language Models Can Teach Themselves to Use Tools," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 36, 2023.
- [14] Pan et al., "Automatically Correcting Large Language Models: Surveying the Landscape of Diverse Self-Correction Strategies," arXiv preprint arXiv:2308.03188, 2023.
- [15] G. Li et al., "CAMEL: Communicative Agents for 'Mind' Exploration of Large Language Model Society," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 36, 2023.
- [16] Z. Gong et al., "MindAgent: Emergent Gaming Interaction," arXiv preprint arXiv:2309.09971, 2023. (
- [17] Y. Jiang et al., "Self-Correction for Distractor-Resilient Multi-hop Question Answering," arXiv preprint arXiv:2106.00373, 2021.
- [18] T. Khot et al., "Decomposed Prompting: A Modular Approach for Solving Complex Tasks," in Proc. Int. Conf. Learn. Represent. (ICLR), 2023.

6. 부록

부록 A. 추론 경로에 대한 정성적 분석

본 부록에서는 HotpotQA 데이터셋의 주요 오류 유형에 대하여, 베이스라인(ReAct) 모델과 제안하는 멀티 에이전트 시스템의 실제 추론 과정을 비교 분석한다.

A.1 환각 오류의 동적 교정

ReAct 모델이 잘못된 검색 결과에 의존하여 환각(Hallucination)을 일으키는 경우, 제안 모델이 이를 어떻게 감지하고 교정하는지 보여준다.

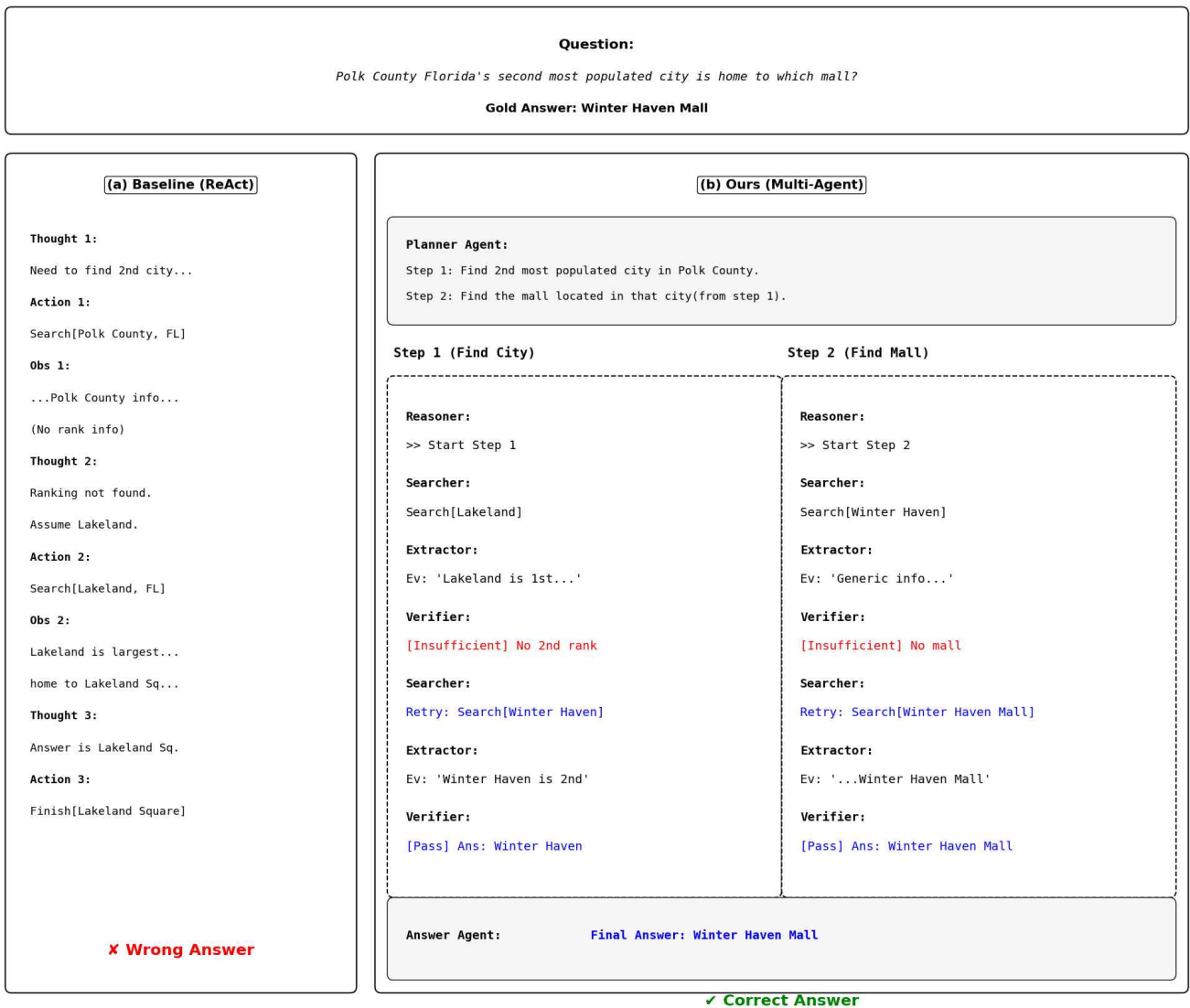


Figure 3. HotpotQA 다중 홉 추론 예제에서의 추론 경로 비교

(a) ReAct 모델은 순위 정보("두 번째로 인구 많은")를 찾는 데 실패하자, 가장 유명한 개체(Lakeland)를 기반으로 환각된 오답을 생성한다. (b) 제안하는 멀티 에이전트 시스템은 문제를 성공적으로 해결한다. Planner가 질문을 분해하고, Reasoner의 자기 검증 메커니즘이 불충분한 증거(빨간색 표시)를 감지하여 동적 교정(Retry)을 트리거함으로써 올바른 개체(Winter Haven)를 찾아내고, 최종적으로 정답(Winter Haven Mall)을 도출한다.

A.2 다단계 정보 연결

복잡한 질문을 해결하기 위해 순차적으로 정보를 검색하고 연결해야 하는 경우, 두 모델의 성능 차이를 비교한다.

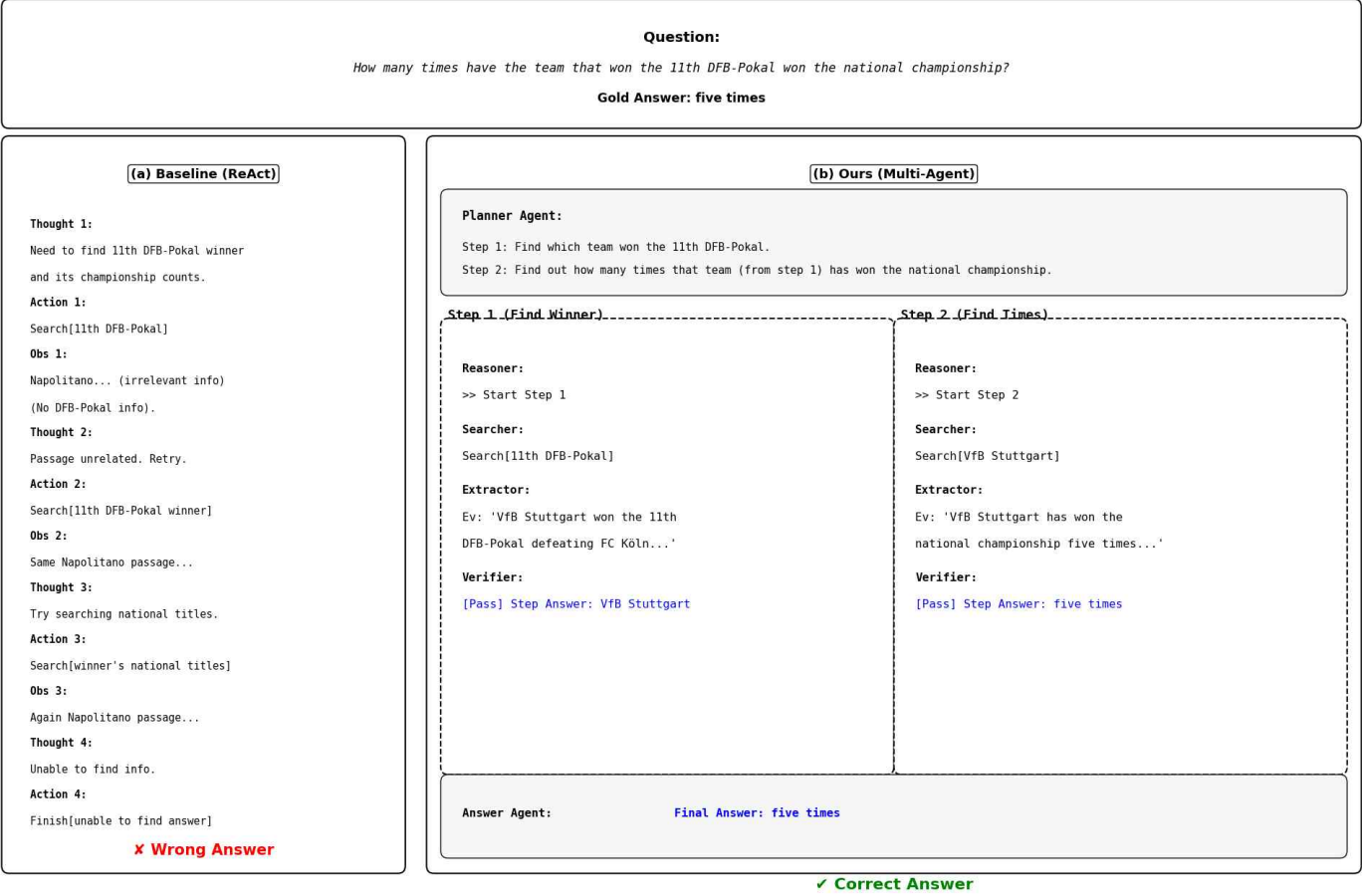


Figure 4. 순차적 정보 검색이 필요한 다단계 질문의 추론 경로 비교

(a) ReAct 모델은 관련 없는 검색 결과로 인해 특정 개체("제11회 DFB-포칼 우승팀")를 찾는 데 실패하고, 답변이 불가하다고 판단한다. (b) 제안하는 멀티 에이전트 시스템은 복잡한 질문을 체계적으로 분해한다. Planner는 작업을 우승팀 식별(Step 1)과 해당 팀의 리그 우승 횟수 집계(Step 2)로 나눈다. Reasoner는 추출된 증거를 검증하며 각 단계를 성공적으로 수행하고, 올바른 개체(VfB Stuttgart)를 다음 단계로 전달하여 최종 정답("five times")을 도출한다.

A.3 불완전 정보를 활용한 논리적 추론

비교 대상 중 일부 정보가 부재할 때, 시스템이 어떻게 대처하는지 보여준다.

