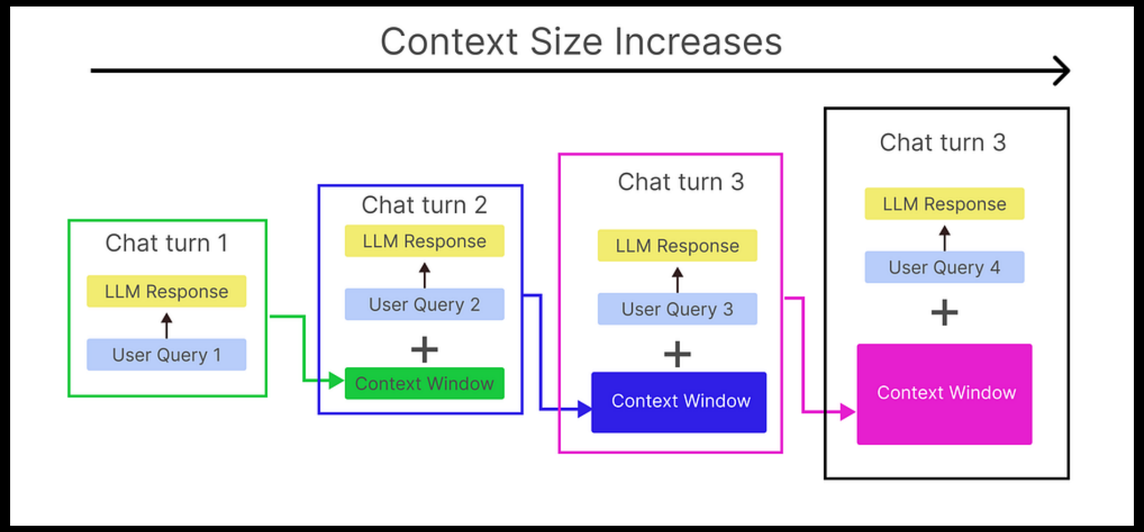
5. Agent 메모리 아키텍처 심화

Agent는 어떻게 기억하는가?

- 메모리의 본질
  - 단순한 정보 저장을 넘어, Agent를 상태(State) 를 가진 존재로 만드는 핵심 요소
  - 과거의 상호작용을 바탕으로 현재를 이해하고 미래의 행동을 최적화하는 능력의 근간
  - 그러나 Transformer 기반 LLM은 메모리 기능이 없음(Memoryless Network)
    - 그저 과거의 모든 텍스트를 동시에 입력받는 방식으로 작동



https://levelup.gitconnected.com/cutting-tokens-by-40-to-lower-llm-api-costs-using-a-memory-efficient-algorithm-e3da975c0424

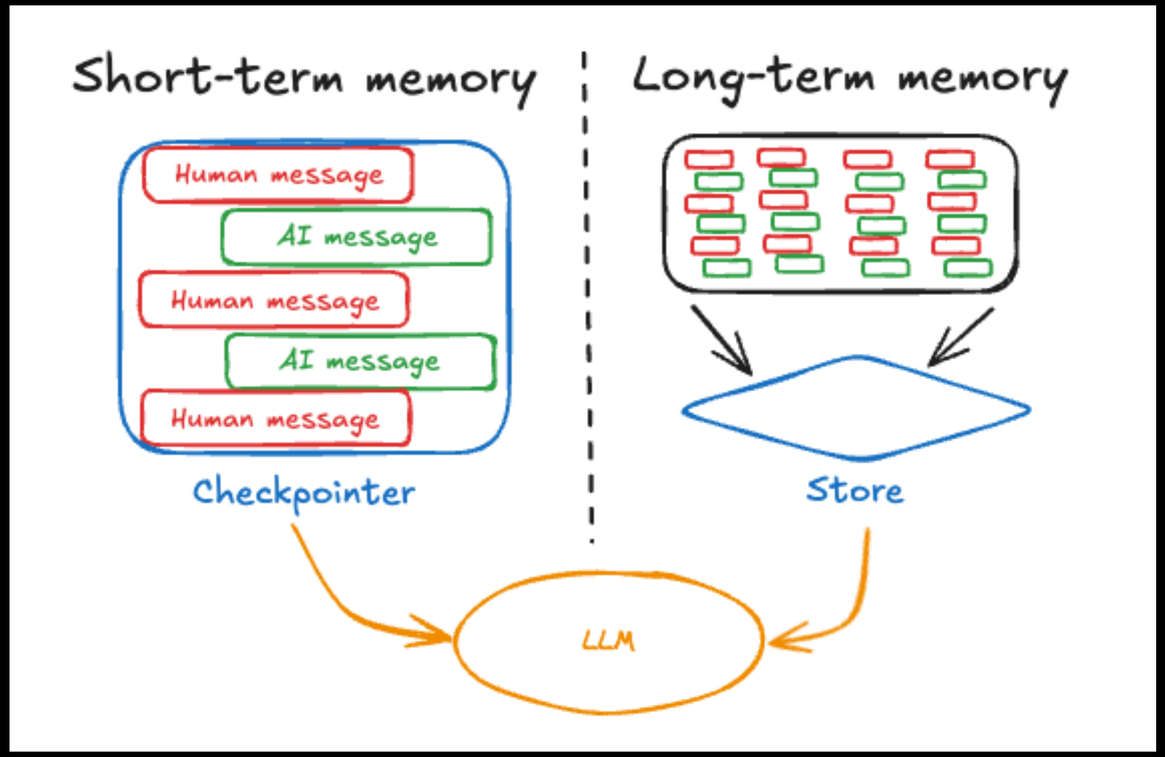
Agent는 어떻게 기억하는가?

- Agent 메모리의 목표
  - 맥락 유지대화의 흐름과 핵심 정보의 지속적 파악
  - 개인화(Personalization): 사용자 선호도 및 과거 이력에 기반한 맞춤형 상호작용
  - 지속적 학습: 과거의 성공과 실패 경험으로부터 점진적으로 성능을 개선하는 자기 성장 능력
- 메모리는 Agent를 일회성 도구에서 신뢰할 수 있는 장기적 Workforce로 진화시키는 동력

Agent Memories

Agent에서의 메모리 유형

- 단기 메모리(Short-Term Memory)
  - 단일 대화 스레드 내의 상호작용
- 장기 메모리(Long-Term Memory)
  - 여러 대화 스레드에 걸쳐 정보를 저장하거나 공유



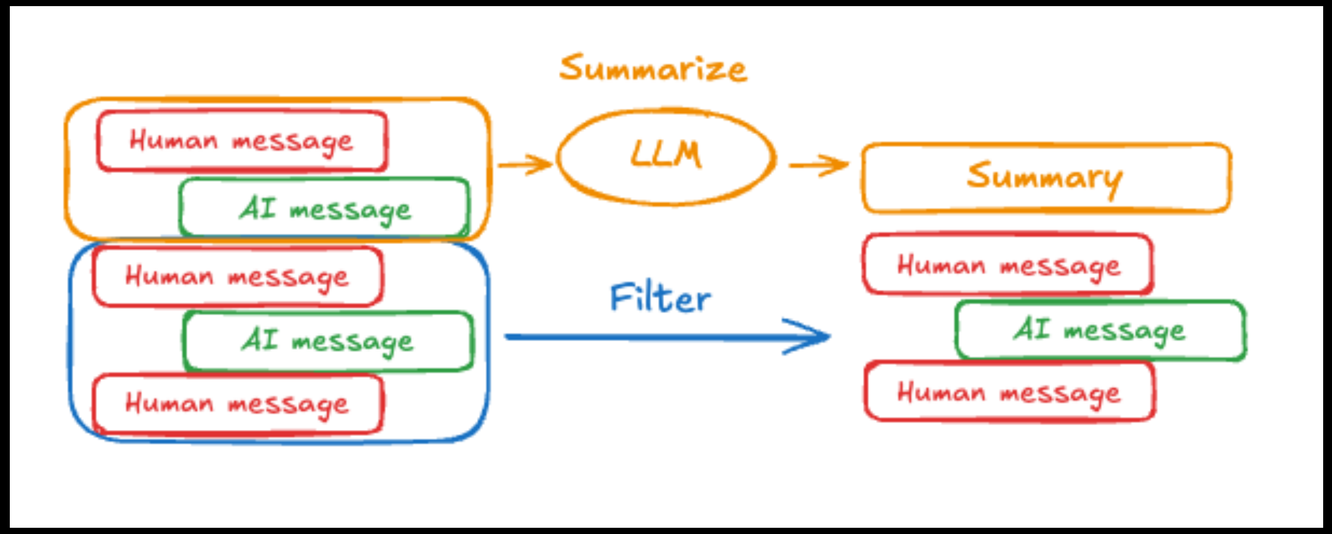
https://langchain-ai.github.io/langgraph/concepts/memory/#collection

Agent Memories

- 단기 기억 (Short-term Memory)
  - 개념
    - 현재 대화의 맥락(Context) 그 자체
    - LLM의 Context Window 내에서 관리되는 Agent의 작업 기억
    - 인간의 RAM과 유사하게, 현재 처리 중인 정보를 빠르고 효율적으로 접근하기 위한 휘발성 공간
  - 특징
    - LLM이 직접 접근 가능하여 즉각적인 맥락 파악에 유리
    - 대화 세션 종료 또는 컨텍스트 창의 한계 초과 시 정보 휘발

Agent Memories

- 단기 기억 (Short-term Memory)
  - 주의 사항
    - 대화가 길어질수록 오래된 정보를 필연적으로 잊어버리는 기억 상실 현상 발생
    - 컨텍스트 창이 클수록 API 호출 비용과 응답 지연 시간(Latency)이 기하급수적으로 증가하는 엔지니어링 Trade-off 존재
    - 이를 위해 요약 메커니즘이 필요



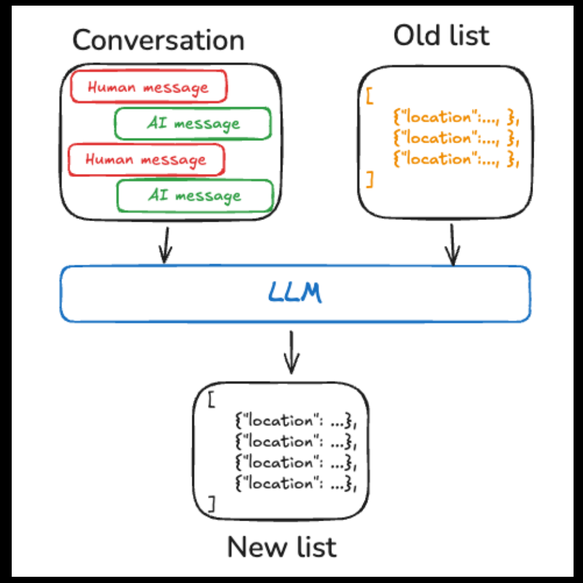
https://langchain-ai.github.io/langgraph/concepts/memory/#collection

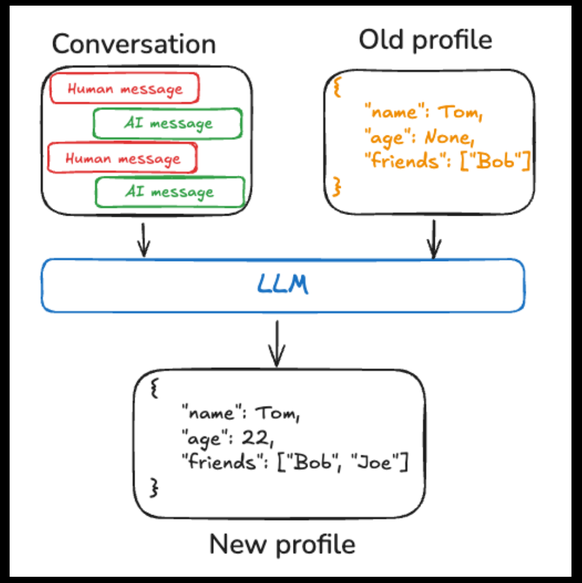
Agent Memories

- 장기 기억/의미 기억 (Long-term Semantic Memory)
  - 개념
    - 대화가 끝나도 유지되는 Agent의 영구적인 지식 베이스(Knowledge Base)
    - 작업을 수행하기 위한 일반적인 사실과 이용자에 대한 정보
    - 데이터, 문서 등 Agent의 활동에 대한 기억
    - 오류 및 디버깅에 대한 기억

Agent Memories

- 장기 기억/의미 기억 (Long-term Semantic Memory)
  - 개념
    - 대화가 끝나도 유지되는 Agent의 영구적인 지식 베이스(Knowledge Base)
    - 작업을 수행하기 위한 일반적인 사실과 이용자에 대한 정보
    - 데이터, 문서 등 Agent가 접근할 수 있는 자료들





Agent Memories

- 장기 기억 - 의미 기억 (Semantic Memory)
  - 기술적인 접근 방법: RAG
    - 인덱싱 (Indexing)
      - 정보를 의미 있는 단위(Chunk)로 분할하고, 각 단위를 벡터 임베딩으로 변환하여 벡터 DB에 저장
      - 문서, 지식, 경험 등
    - 검색 (Retrieval)
      - 사용자 질문이나 다른 에이전트의 들어오면, 벡터 DB에서 의미적으로 가장 유사한 정보 조각들을 검색
    - 증강 (Augmentation)
      - 검색된 정보 조각들을 LLM의 컨텍스트에 주입하여, Agent의 액션 타임을 줄일 수 있음

Agent Memories

- 장기 기억 - 일화 기억 (Episodic Memory)
  - 단순히 정보를 저장하는 것을 넘어, 특정 작업의 성공 또는 실패 경험을 저장
    - Agent의 자서전적 기억(Autobiographical Memory)
    - "나는 과거에 이렇게 행동했다"와 같이 어떻게 문제를 해결했는지에 대한 절차적 지식
  - 각 에피소드를 구조화된 데이터로 저장하여 학습 효율 극대화 및 시행착오 단축
    - 과거의 성공적인 문제 해결 절차를 학습하고 유사한 미래 문제에 재사용

```
{
  "task_goal": "신규 AI 모델에 대한 기술 블로그 초안 작성",
  "steps_taken": [
    {"action": "web_search", "input": "최신 AI 모델 동향"},
    {"action": "summarize", "input": "검색 결과 요약"},
    {"action": "write_draft", "input": "요약본 기반 초안 작성"}
  ],
  "final_outcome": "success",
  "user_feedback": "전문 용어 설명이 추가되면 좋겠음"
}
```

기억의 한계

- 기억의 한계 - 점과 선의 문제
  - 지금까지의 기억 방식이 가진 근본적 한계
  - 의미 기억 (RAG)
    - 문제와 관련된 정보 점들을 흩뿌려진 형태로 찾아줌 (예: '회사 A'에 대한 문서, '인물 B'에 대한 문서)
  - 일화 기억
    - 특정 목표를 향한 행동의 순차적 경로를 제공함 (예: 보고서 작성 절차)

기억의 한계

- 기억의 한계 - 점과 선의 문제
  - 해결되지 않은 문제
    - 흩어진 정보 점들 사이의 관계는 어떻게 파악하는가
    - RAG는 '회사 A'와 '인물 B'가 관련 있다는 것은 알지만, '인물 B가 회사 A의 창업자'라는 구조적 관계는 명시적으로 알려주지 못함
    - 이는 Agent가 피상적인 정보 나열을 넘어, 깊이 있는 추론을 하는 데 있어 결정적인 장벽으로 작용

하이브리드 접근법

- 하이브리드 접근법 (RAG + Knowledge Graph)
  - RAG의 비정형 텍스트 검색 능력
  - 지식 그래프(Knowledge Graph, KG)의 정형적 관계 추론 능력
  - 위 둘을 결합하여 기억의 깊이와 정확성을 극대화하는 방식
    - RAG (점 찾기)
      - 질문과 관련된 비정형 텍스트를 신속하게 검색하여 풍부한 서술형 정보 제공
    - Knowledge Graph (선 잇기)
      - 세상의 지식을 노드(개체)와 간선(관계)으로 구성된 그래프 형태로 저장하여
      - 정보 간의 구조적, 인과적 관계를 표현
      - 예: <데미스 하사비스> -[CEO_OF]-> <딥마인드> -[ACQUIRED_BY]-> <구글>

하이브리드 접근법

- 하이브리드 접근법 (RAG + Knowledge Graph) 작동 방식
  - Query: "알파벳이 딥마인드를 인수한 후, 데미스 하사비스의 역할은 어떻게 변했나?”
  - 1단계: RAG 검색 (점 찾기)
    - Agent가 RAG 시스템을 통해 관련 뉴스 기사, 인터뷰, 공식 발표문 등 다수의 텍스트 문서를 검색
  - 2단계: KG 조회 (선 잇기)
    - Agent가 내부 지식 그래프에서 알파벳, 딥마인드, 데미스 하사비스 노드를 조회
    - <딥마인드>-[ACQUIRED_BY]-><알파벳> 이라는 명확한 인수 관계와 <데미스 하사비스>-[CEO_OF]-><딥마인드> 라는 직책 관계를 즉시 파악

하이브리드 접근법

- 하이브리드 접근법 (RAG + Knowledge Graph) 작동 방식
  - Query: "알파벳이 딥마인드를 인수한 후, 데미스 하사비스의 역할은 어떻게 변했나?”
  - 3단계: 종합 추론 (점과 선의 결합)
    - Agent는 KG를 통해 '누가 누구를 인수했고, 누가 어디 소속인지'라는 구조적 뼈대를 이해
    - 이 뼈대 위에 RAG가 찾아온 텍스트 정보("...그는 이후 구글의 통합 AI 부문을 이끌게 되었다...")라는 서술적 살을 붙여 최종 답변 생성
    - 결과: "알파벳의 딥마인드 인수 이후, CEO였던 데미스 하사비스는 구글의 통합 AI 부문을 총괄하는 역할로 확장되었습니다" 와 같은 깊이 있는 답변 가능