Stable Planning in LLM Agents

구선민

Natural Language Processing
& Artificial Intelligence

Coarse-to-Fine Grounded Memory for LLM Agent Planning

Wei Yang[1], Jinwei Xiao[1], Hongming Zhang[1†], Qingyang Zhang[1], Yanna Wang[1], Bo Xu[1†]
[1]National Key Laboratory of Cognition and Decision Intelligence for Complex Systems,
Institution of Automation, Chinese Academy of Sciences
{yangwei2023,xiaojinwei2024,hongming.zhang}@ia.ac.cn
{zhangqingyang2019,wangyanna2013,boxu}@ia.ac.cn
[†]**Correspondence:** hongming.zhang@ia.ac.cn, boxu@ia.ac.cn

Introduction

Motivation

- 에이전트는 종종 비효율적인 탐색으로 관련 없거나 부족한 데이터를 수집함
- partially observable environments 에서 추론 이탈로 어려움을 겪는데, 불완전한 observations은 연쇄적인 오류를 유발함
- partially observable environments?
  - 에이전트가 환경의 전체 상태를 한 번에 또는 완벽하게 인지할 수 없는 상황
  - e.g., , 에이전트가 앞은 볼 수 있지만 뒤나 벽 너머를 볼 수 없는 경우

Introduction

Motivation

- 추론 이탈 문제
  - **Incomplete observations**: 에이전트가 환경의 중요한 부분을 보지 못하거나, 과거의 중요한 정보를 기억하지 못하여 현재 상황을 정확하게 이해하지 못함
  - **Cascading errors**: incomplete observations 로 인해 잘못된 판단을 내리면 이 잘못된 판단이 이후의 행동과 관측에 영향을 미쳐 연속적인 오류를 발생시킬 수 있음

Introduction

Motivation

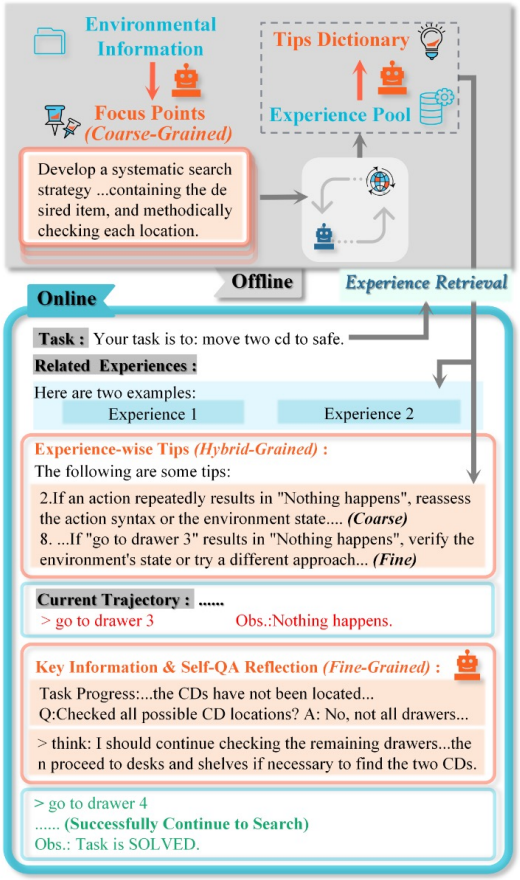
- 최근 연구들은 동적 환경 상호작용에서 single granularity memory 에 의존하며 이는 수집된 경험의 품질에 의해 효과가 제한됨
- 이러한 한계는 지식의 다양성과 계획의 유연성을 저해하며 구체적으로 세 가지 문제로 나타남
    1. 어려운 플래닝 태스크에서 LLM의 초기 탐색 효율성이 최적화되지 않아 메모리 품질이 제한
    2. 지식 추출이 전체적인 수준이나 특정 시나리오 수준에서 이루어져 획득된 지식의 세분화가 획일적
    3. Manually designed question 을 통한 reflection 은 일반화 능력이 부족함

→ LLM의 내부 지식을 활용하여 메모리 수집 및 활용을 가이드 함으로써 복잡한 환경에서 플래닝 능력을 향상시키는 새로운 에이전트 프레임워크인 Coarse-to-Fine Grounded Memory (CFGM) 제안

Methods

Coarse-to-Fine Grounded Memory (CFGM)

- CFGM 은 세 가지 점진적인 메모리 그라운딩 단계 포함
  1. Coarse-grained focus points
  2. Hybrid-grained tips
  3. Fine-grained key information analysis & self-QA reflection



Methods

Coarse-Grained Focus-Driven Experience Collection

- 다양한 경험적 메모리를 수집하기 위해 reflection 채택해서 collaborative thinking 과 retry-reflection
- LLM이 environmental foundational information 도출하여 LLM을 가이드하는 focus point 를 얻도록 해서 환경에 대한 자체적인 사전 이해를 바탕으로 더 높은 품질의 경험을 수집하도록 함

Algorithm 1 Coarse-Grained Focus-Driven Experience Collection.

$N = len(\mathcal{T}_{\text{train}}), n = 0$
$FP = \text{LLM}_{\text{Focus}}(Desc_{\text{env}}, F_{\text{manual}})$
while task $n < N$ **do**
  $t_n \leftarrow \mathcal{T}_{\text{train}}[n], \nu_{n,0} \leftarrow$ ""
  for trial $z = 0$ to Z **do**
    Initialize Trajectory $\tau_{n,z} \leftarrow$ env.reset(t_n)
    for timestep $i = 0$ to H **do**
      $a_i \leftarrow \text{LLM}_{\text{ReAct}}(a_i \mid \tau_{n,z}, F_{\text{manual}}, \nu_{n,z}, FP)$
      o_{i+1}, done $\leftarrow$ env.step(a_i)
      $\tau_{n,z} \leftarrow \tau_{n,z} \cup \{(a_i, o_{i+1})\}$
      if done **then**
        break
      end if
    end for
    $\mathcal{B} \leftarrow \mathcal{B} \cup (t_n, \tau_{n,z})$
    if done or $z = Z$ **then**
      $n \leftarrow n + 1$
      break
    else
      $\nu_{n,z+1} \leftarrow \text{concat}(\nu_{n,z} + \text{LLM}_{\text{Reflect}}(\tau_{n,z}))$
    end if
  end for
end while
return $\mathcal{B}$

성공 여부와 관계 없이 모든 궤적 저장

실패원인 분석해서 다음 시도 때 활용

Methods

Hybrid-Grained Experience-Wise Tips Extraction

- 이전 단계는 trial-and-error paradigm 로 경험 풀에 저장된 태스크에는 성공 및 실패 궤적이 모두 존재함
- 이러한 궤적들은 fine-grained scene-specific + 유사한 태스크 전반에 걸쳐 적용 가능한 high-level coarse-grained 인사이트 포함

  → 이러한 하이브리드 수준의 지식을 팁이라고 부름
- 모델의 내부 지식을 오프라인으로 활용하여 주어진 태스크 경험의 여러 궤적을 하이브리드 수준의 관점을 혼합한 set of tips로 변환

Methods

Hybrid-Grained Experience-Wise Tips Extraction

- 경험 풀 B에 있는 모든 궤적 데이터를 태스크 t_n 에 따라 re-indexing
- successful and failed attempts 를 모두 포함하는 궤적은 $C_{compare}$ 에 담고, successful attempt 만은 $C_{success}$ 에 담음
- 모든 training task t_n 를 반복
  - 만약 t_n 이 $C_{compare}$에 나타나면, LLM_{Tips}에 해당 태스크의 성공적 궤적과 실패 궤적을 비교하도록 프롬프트하고 오류를 방지하는 데 도움이 되는 팁 추출
  - LLM에게 성공 궤적에만 집중하여 추가적인 성공별 요인을 개선하고 팁을 확장하도록 요청
- 결과로 얻어진 팁은 Tips Dictionary TD 에 저장

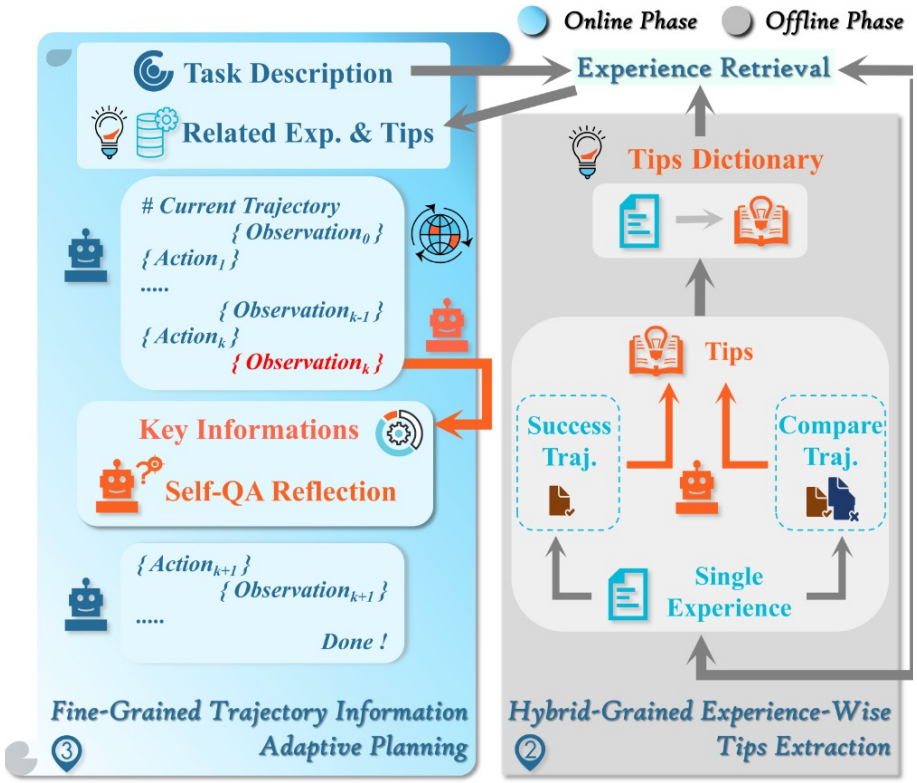
Algorithm 2 Hybrid-Grained Experience-Wise Tips Extraction.

$N = len(\mathcal{T}_{\text{train}})$
Construct fail/success tuples of the same tasks in $\mathcal{B}$:
$C_{\text{compare}} = \{t_1 : (\tau_1^{\text{success}}, \tau_{1,0}^{\text{fail}}, \ldots), \ldots,$
$\quad t_i : (\tau_i^{\text{success}}, \tau_{i,0}^{\text{fail}}, \ldots), \ldots\}$
Construct success trajectories in $\mathcal{B}$:
$C_{\text{success}} = \{t_1 : \tau_1^{\text{success}}, \ldots, t_i : \tau_i^{\text{success}}, \ldots\}$
for task $m = 0$ to $N - 1$ **do**
  $t_n \leftarrow \mathcal{T}_{\text{train}}[n]$
  if $t_n \in C_{\text{compare}}$ **then**
    $TD[t_n] = \text{LLM}_{\text{Tips}}(C_{\text{compare}}[t_n])$
    $TD[t_n] \leftarrow (TD[t_n],$
      $\text{LLM}_{\text{Tips}}(C_{\text{success}}[t_n], TD[t_n]))$
    continue
  end if
  if $t_n \in C_{\text{success}}$ **then**
    $TD[t_n] = \text{LLM}_{\text{Tips}}(C_{\text{success}}[t_n]))$
  end if
end for
return TD

Methods

Fine-Grained Trajectory Information Adaptive planning

- 에이전트가 온라인 환경에서 복잡한 태스크를 수행할 때 기존의 경험과 새로운 정보를 활용하여 유연하게 계획을 조정하고 오류를 수정하도록 도움
- 오프라인에서 수집된 경험과 추출된 hybrid-grained knowledge(팁)을 온라인 추론 과정에서 동적으로 활용
- 예상치 못한 상황이나 환경 이상에 직면했을 때, LLM이 현재 궤적 상태를 fine-grained 분석하여 planning derailment 을 완화하고 적응적으로 수정 계획을 세우도록 함



Experiments

Experimental Settings

Environments.

- AlfWorld: Virtual household agent benchmark
- WebShop: Interactive environment simulating online shopping
- ScienceWorld: Text-based environment focusing on elementary

Evaluation.

- Success Rate (SR)
  - AlfWorld: 시간 제한 내 태스크 완료했는지
  - WebShop: 속상에 맞는 물건 구매했는지
  - ScienceWorld: 태스크 완료 여부

Experiments

Main Results

- CFGM 의 효과성 및 다양한 환경에서 강건성 확인

Method	Features			AlfWorld	WebShop		ScienceWorld
	Off. Exp.	Traj. Ref.	Gro. Mem.	Success Rate (SR)↑	Reward↑	SR↑	SR↑
ReAct	✗	✗	✗	80.60% ± 0.68%	58.6 ± 1.0	37% ± 2%	43% ± 1%
ExpeL	✓	✗	✗	81.34% ± 0.85%	62.2 ± 1.3	42% ± 3%	57% ± 2%
AutoGuide	✓	✗	✗	83.58% ± 0.77%	73.3 ± 1.4	47% ± 2%	N/A
QuBE	✗	✓	✗	84.33% ± 0.73%	N/A	N/A	N/A
ExpeL + QuBE	✓	✓	✗	<u>85.07% ± 1.03%</u>	<u>75.6 ± 1.3</u>	<u>49% ± 3%</u>	<u>65% ± 2%</u>
CFGM	✓	✓	✓	**91.00% ± 0.82%**	**85.0 ± 1.3**	**57% ± 3%**	**74% ± 2%**

Table 1: Main results across AlfWorld, WebShop, and ScienceWorld benchmarks. Off. Exp. refers to collecting offline experiences, Traj. Ref. refers to reflection based on trajectories during online inference, and Gro. Mem. refers to leveraging grounded memory with LLM's knowledge. Best in bold, second-best underlined. CFGM consistently outperforms the baselines on all domains, highlighting the importance of comprehensive memory grounding and the scalability of CFGM in diverse environments.

Experiments

Ablation Studies

- **Effectiveness of Different Components**

FP	✗	✓	✗	✗	✗	✓
ET	✗	✗	✓	✗	✓	✓
KIR	✗	✗	✗	✓	✓	✓
SR%	80.60	85.82	86.57	85.82	88.06	91.00

Table 2: Module ablation experiment results on AlfWorld. FP refers to using focus points to guide experience collection, ET refers to extracting experience-wise tips, and KIR refers to question-answer self-reflection based on key information during online planning.

- **Generalizability across Different Models.**

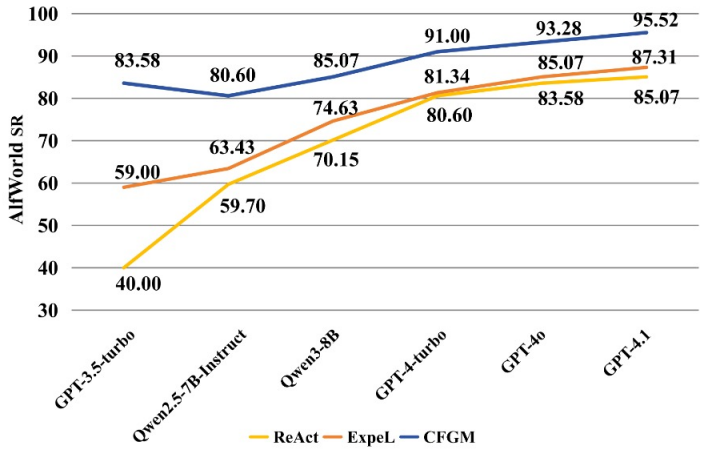


Figure 3: The SR achieved by different methods using various models on AlfWorld. CFGM demonstrates strong generalization across different models and consistently outperforms the baselines.

Experiments

Ablation Studies

- **Impact of Experience Retrieval Scope on Tips Quality.**

k	0	1	2	3	5
SR	80.60%	82.09%	86.57%	86.57%	81.34%

Table 3: The impact of different Top-k experience retrieval scope on the quality of tips on AlfWorld. The retrieved top-2 experiences yield tip sets with optimal quality and retrieval efficiency.

- **Experience Tips' Generalizability to Out-of-Distribution Environments.**

Method	WebArena–Shopping SR
ReAct	10.2% ± 0.5%
ExpeL	18.5% ± 0.9%
AutoGuide	20.4% ± 0.7%
CFGM	25.1% ± 0.8%

Table 4: Out-of-distribution generalization of different method's extracted experience knowledge from WebShop on the 98 WebArena–Shopping tasks. hybrid-grained tips of CFGM perform superior out-of-domain generalization compared to existing knowledge extraction agents after transfer.

Experiments

Ablation Studies

- **Comparative Analysis of Online Reflection.**

Method	AlfWorld SR
Self-QA Reflection	84.33%
QA Reflection(Fixed Question)	84.33%
Key Information Reflection	85.82%

Table 5: The comparative analysis of different reflection. Fine-grained grounded memory enables more flexible and effective reflection than manual QA approaches. Prior short-term memory analysis (key information) further enhances comprehension and reasoning when leveraging grounded memory.

- **The Overall Efficiency of CFGM.**

Method	Off. Tokens	On. Tokens	On. Turns
ReAct	-	1734.6	19.01
ExpeL	3888.2	5125.5	17.32
AutoGuide	4873.4	4809.4	16.55
QuBE	-	4752.3	17.21
CFGM	4068.5	4681.4	14.32

Table 6: Average token consumption for processing a single training data sample during the offline phase on AlfWorld (Off. Tokens). Average token consumption per step and average interaction turns per complete trajectory during online inference on AlfWorld (On. Tokens and On. Turns).

Memory OS of AI Agent

Jiazheng Kang
Beijing University of Posts
and Telecommunications
kjz@bupt.edu.cn

Mingming Ji
Tencent AI Lab
matthhewj@tencent.com

Zhe Zhao
Tencent AI Lab
nlpzhezhao@tencent.com

Ting Bai *
Beijing University of Posts
and Telecommunications
baiting@bupt.edu.cn

Introduction

Motivation

- 고정된 컨텍스트 윈도우와 불충분한 메모리 관리로 인해 대화 연속성 유지 실패, 사실 불일치, 개인화 부족 문제
- 일관된 페르소나 유지를 위해서는 long-term memory 일관성이 매우 중요
- 기존 메모리 관리 방법론들은 저장 구조, 검색 메커니즘, 업데이트 전략 등 단일 차원에만 포커스
- AI 에이전트를 위한 체계적이고 포괄적인 메모리 관리를 가능하게 하는 unified operating system 제안되지 않았음

→ 운영 체제의 메모리 관리 원칙에서 영감을 받아, AI 에이전트를 위한 포괄적인 메모리 운영 체제인 MemoryOS 제안

Methods

Overview

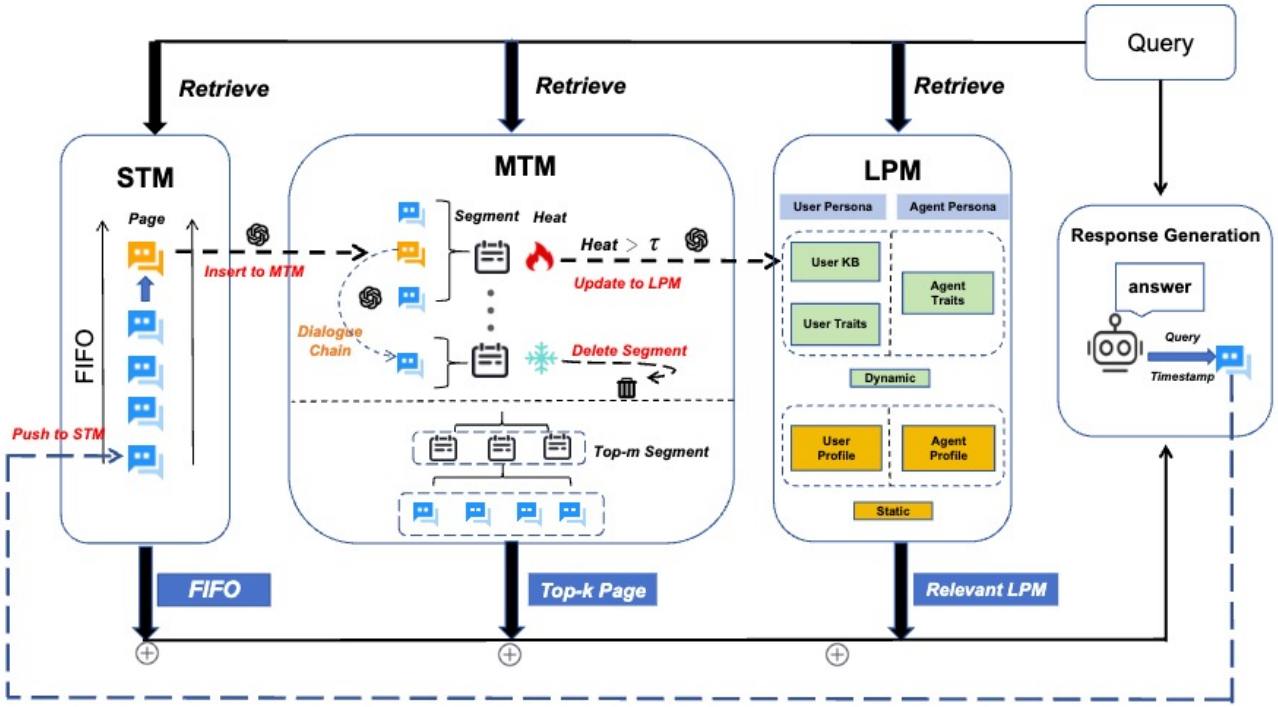


Figure 1: The overview architecture of MemoryOS, including memory Store, Updating, Retrieval, Response.

Methods

Memory Storage Module

- 세 가지 계층적 저장 단위를 통해 메모리를 조직하고 저장
- **1) Short-Term Memory (STM):** 현재 진행 중인 실시간 대화 데이터를 dialogue pages 단위로 저장
  - 각 페이지는 사용자 질의(Q), 모델 응답(R), 타임스탬프(T)로 구성

$$page_i = \{Q_i, R_i, T_i\}$$

  - Dialogue Chain: 각 페이지에는 컨텍스트 정보 $page_i^{chain}$ 추가
- **2) Mid-Term Memory (MTM):** OS 메모리 관리 원칙에서 영감을 받은 Segmented Paging 저장 아키텍처
  - 동일한 주제를 가진 대화 페이지들은 세그먼트로 그룹화
  - 세그먼트 내용은 관련 대화 페이지를 기반으로 LLM에 의해 요약
  - 의미론적 및 키워드 유사성을 기반으로 대화 페이지와 세그먼트 간의 유사성을 측정

Methods

Memory Storage Module

- 세 가지 계층적 저장 단위를 통해 메모리를 조직하고 저장
- **3) Long-term Persona Memory (LPM):** user 및 assistant가 중요한 개인 정보 및 특성을 지속적으로 기억하여 long-term 상호작용에서 일관성 및 개인화 보장
  - User Persona
    - Static component: 성별, 이름, 생년월일
    - User KB: 과거 상호작용에서 추출된 factual information 을 동적 저장
    - User Traits: 사용자의 관심사, 습관 및 선호도 같은 성향이나 행동적 특성
  - Agent Persona
    - Agent Profile: AI 에이전트가 수행하는 역할이나 성격 특성과 같은 고정된 세팅
    - Agent Traits: user와 상호작용을 통해 발전하는 동적 속성

Methods

Memory Update Module

STM-MTM Update (FIFO 방식)

- STM은 dialogue pages 형태로 실시간 대화 데이터를 고정 길이의 큐에 저장
- 새로운 대화 페이지는 큐의 끝에 추가
- STM 큐가 최대 용량에 도달하면 가장 오래된 대화 페이지는 FIFO(First-In-First-Out) 에 따라 STM에서 MTM으로 전송

Methods

Memory Update Module

MTM-LPM Update (Heat 점수 기반)

- segment deletion 및 segment-to-LPM updates

$$Heat = \alpha \cdot N_{visit} + \beta \cdot L_{interaction} + \gamma \cdot R_{recency}$$

- N_{visit} : 세그먼트가 검색된 횟수
- $L_{interaction}$: 세그먼트 내의 총 대화 페이지 수
- $R_{recency}$: time decay 계수
- 세그먼트의 길이가 최대 용량을 초과할 경우 가장 낮은 Heat 점수를 가진 세그먼트 제거

Methods

Memory Update Module

LPM Update

- LPM 으로 전송된 세그먼트 및 대화 페이지는 User Traits, User KB, Agent Traits 업데이트
- User Traits는 3가지 카테고리(basic needs and personality, AI alignment dimensions, and content platform interest tags), 90 demensions 으로 구성되며 LLM에 의해 자율적으로 특성이 진화하도록 추출 및 업데이트됨
- 사용자 및 에이전트 어시스턴트와 관련된 factual information은 각각 User KB와 Agent Traits에 추출되어 기록

Methods

Memory Retrieval Module

- 사용자 쿼리에 대한 응답을 생성하기 위해 저장된 메모리에서 관련 정보를 효율적으로 검색하는 역할
- STM, MTM, LPM에서 정보 검색
- STM검색: 현재 대화의 가장 최근 컨텍스트를 담고 있기 때문에 모든 대화 페이지 검색
- MTM 검색:
  - 사용자 쿼리와의 매칭 점수(Fscore)를 사용하여 가장 관련성이 높은 상위 m개 후보 세그먼트 선택
  - 선택된 세그먼트 내에서 의미론적 유사성을 기반으로 가장 관련성이 높은 상위 k개의 대화 페이지 검색
  - 검색 후에는 N_{visit} 및 $R_{recency}$ 가 업데이트 되어 Heat Score에 반영
- LPM 검색: 사용자 쿼리 벡터에 대한 의미론적 관련성을 고려해서 top-10 검색

Methods

Response Generation Module

- 사용자 쿼리가 주어지면, STM, MTM, LPM에서 검색된 세 가지 유형의 콘텐츠 합쳐서 최종 프롬프트 구성

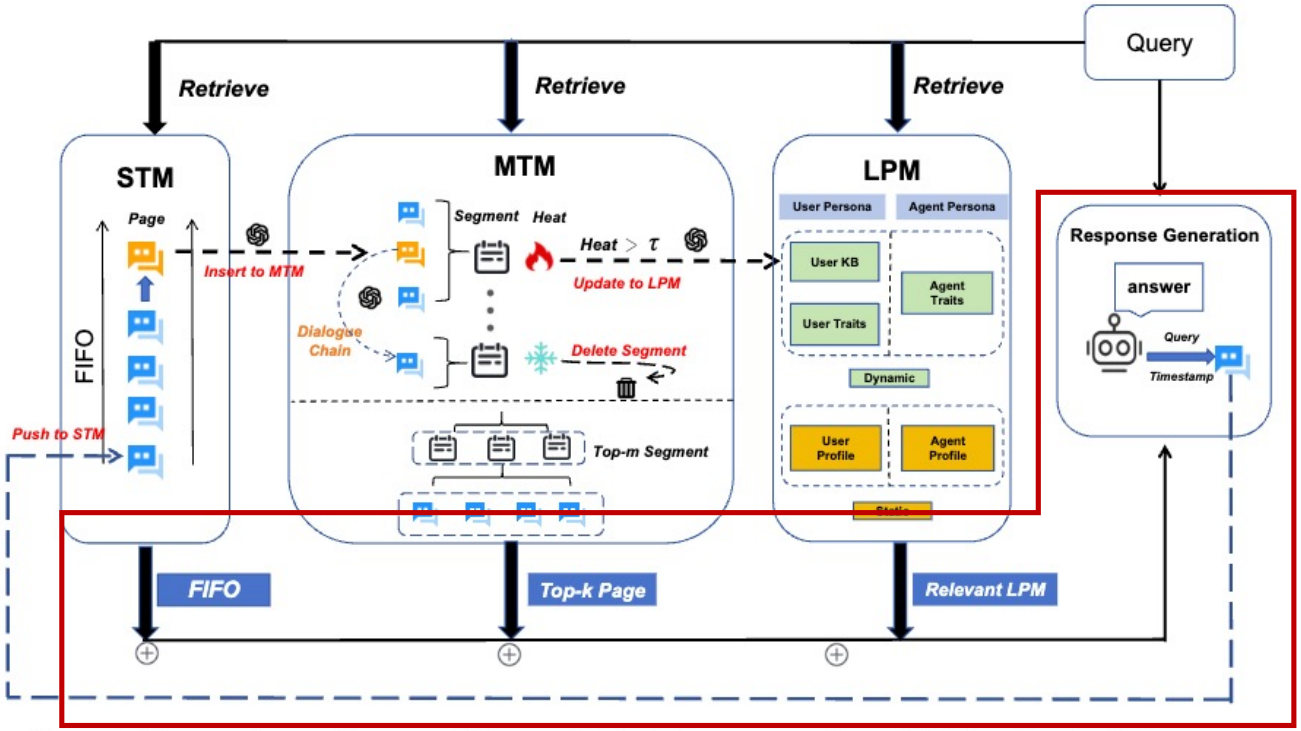


Figure 1: The overview architecture of MemoryOS, including memory Store, Updating, Retrieval, Response.

Experiments

Experimental Settings

Datasets.

- GVD: 15명의 가상 사용자와 AI 에이전트 간의 10일간의 상호작용을 시뮬레이션하여 생성된 멀티턴 대화
- LoCoMo: 평균 300 턴 및 약 9k로 구성된 long-term 대화

Evaluation.

- GVD: Memory Retrieval Accuracy (Acc.), Response Correctness (Corr.), Contextual Coherence (Cohe.)
  - DeepSeek-R1 로 평가
- LoCoMo: F1, BLEU

Experiments

Main Results

- MemoryBank가 가장 낮은 성능 → 단순 memory decay mechanism 적용은 효과적 대화 메모리 관리 X
- A-Mem과 MemGPT 은 상대적으로 long-term 성능 좋지만 체계적인 메모리 관리 메커니즘 부족
- MemoryOS 는 안정적 관리로 높은 성능

Table 1: Comparison results on the GVD dataset.

Model	Method	Acc. ↑	Corr. ↑	Cohe.↑
GPT-4o-mini	TiM	84.5	78.8	90.8
	MemoryBank	78.4	73.3	91.2
	MemGPT	87.9	83.2	89.6
	A-Mem	90.4	86.5	91.4
	Ours	**93.3**	**91.2**	**92.3**
Improvement (%)		3.2% ↑	5.4% ↑	1.0% ↑
Qwen2.5-7B	TiM	82.2	73.2	85.5
	MemoryBank	76.3	70.3	82.7
	MemGPT	85.1	80.2	86.9
	A-Mem	87.2	79.5	87.8
	Ours	**91.8**	**82.3**	**90.5**
Improvement (%)		5.3% ↑	3.5% ↑	3.1% ↑

Table 2: LoCoMo dataset comparison with per-category scores and average ranks. A-Mem refers to the results reported in the original paper. A-Mem* represents our implementation results under the same experimental environment as our model.

Model	Method	Single Hop		Multi Hop		Temporal		Open Domain		Avg. Rank ↓	Avg. Rank ↓
		F1 ↑	BLEU-1 ↑	F1 ↑	BLEU-1 ↑	F1 ↑	BLEU-1 ↑	F1 ↑	BLEU-1 ↑	(F1)	(BLEU-1)
GPT-4o-mini	TiM	16.25	13.12	18.43	17.35	8.35	7.32	23.74	22.05	3.8	4.0
	MemoryBank	5.00	4.77	9.68	6.99	5.56	5.94	6.61	5.16	5.0	5.0
	MemGPT	26.65	17.72	25.52	19.44	9.15	7.44	41.04	34.34	2.2	2.5
	A-Mem	27.02	20.09	45.85	36.67	12.14	12.00	44.65	37.06	–	–
	A-Mem*	22.61	15.25	33.23	29.11	8.04	7.81	34.13	27.73	3.0	2.5
	Ours	**35.27**	**25.22**	**41.15**	**30.76**	**20.02**	**16.52**	**48.62**	**42.99**	**1.0**	**1.0**
Improvement (%)		32.35%↑	42.33%↑	23.83%↑	5.67%↑	118.80%↑	111.52%↑	18.47%↑	25.19%↑	–	–
Qwen2.5-3B	TiM	4.37	5.01	2.54	3.21	6.20	5.37	6.35	7.34	4.3	3.5
	MemoryBank	3.60	3.39	1.72	1.97	6.63	6.58	4.11	3.32	4.8	4.8
	MemGPT	5.07	4.31	2.94	2.95	7.04	7.10	7.26	5.52	2.8	3.8
	A-Mem	12.57	9.01	27.59	25.07	7.12	7.28	17.23	13.12	–	–
	A-Mem*	10.31	8.76	16.31	11.07	6.94	7.31	12.34	10.62	2.3	2.0
	Ours	**23.26**	**15.39**	**21.44**	**14.95**	**10.18**	**8.18**	**26.23**	**22.39**	**1.0**	**1.0**
Improvement (%)		125.61% ↑	75.68% ↑	31.45% ↑	35.05% ↑	46.69% ↑	11.90% ↑	112.56% ↑	110.83% ↑	–	–

Experiments

Main Results

- 모델 효율성을 평가하기 위해 tokens consumed 및 average LLM calls in each response 측정
- MemoryOS는 Top-2 baseline(MemGPT, A-Mem)보다 뛰어난 성능
- A-Mem*에 비해 LLM calls 훨씬 적고(4.9 vs. 13), MemGPT에 비해 토큰 소비가 훨씬 적음(3,874 vs. 16,977)

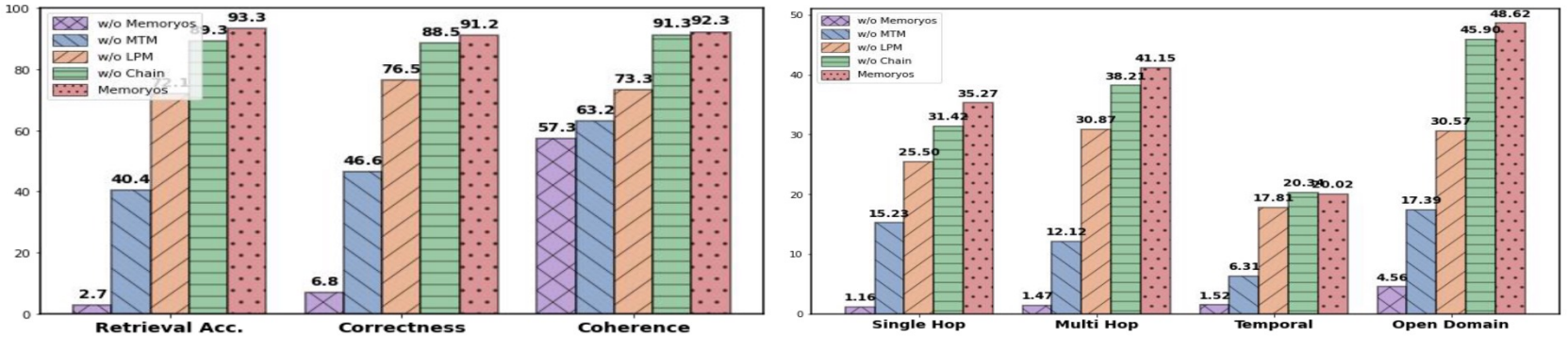
Table 3: Efficiency analysis on LoCoMo benchmark (quantified by LLM call and recalled tokens counts).

Method	Tokens	Avg.Calls	Avg. F1
MemoryBank	432	3.0	6.84
TiM	1,274	2.6	18.01
MemGPT	16,977	4.3	29.13
A-Mem*	2,712	13.0	26.55
Ours	**3,874**	**4.9**	**36.23**

Experiments

Ablation Study

- w/o MTM, w/o LPM. w/o Chain 순서로 성능 하락폭 큼
- 단기적인 컨텍스트 연결 메커니즘이 MTM이나 LPM과 같은 계층적 저장 구조만큼 전체 시스템 성능에 큰 영향 미치지 않음



(a)Ablation results on the GVD dataset with GPT-4o-mini

(b)Ablation results on the LoCoMo dataset with GPT-4o-mini using the F1 score.

Figure 2: The ablation study on the GVD and LoCoMo benchmark datasets.

Experiments

Ablation Study

- MTM에서 검색된 dialogue pages 수가 모델 성능에 미치는 영향 분석
- k가 증가함에 따라 모델 성능이 향상되지만 임계값을 초과하면 효과 감소
- 과도한 내용은 노이즈를 유발하여 성능에 부정적인 영향을 미침

  → k=10 으로 세팅한거 정당화

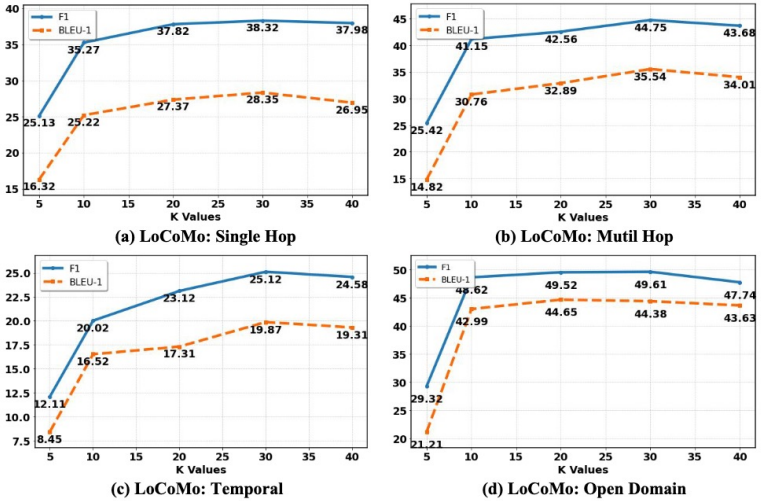


Figure 3: Impact of hyperparameter k (retrieved pages in MTM) on LoCoMo benchmark.

감사합니다