INMS: Memory Sharing for Large Language Model based Agents

Anonymous ACL submission

Abstract

The adaptation of Large Language Model (LLM)-based agents to execute tasks via natural language prompts represents a significant advancement, notably eliminating the need for explicit retraining or fine tuning, but are constrained by the comprehensiveness and diversity of the provided examples, leading to outputs that often diverge significantly from expected results, especially when it comes to the open-ended questions. Although Retrieval-Augmented Generation (RAG) can effectively address this problem, its implementation may be hindered by the scarcity of suitable external databases or the insufficiency and obsolescence of examples in existing databases. This work aims to address the problem of external datasets shortage and obsolescent for databases. We proposed a novel **INteractive Memory Sharing** framework, which integrates the real-time memory filter, storage and retrieval to enhance the In-Context Learning process. This framework allows for the sharing of memories among agents, whereby the interactions and shared memories between agents effectively enhance the diversity of the memories. The collective self-enhancement through interactive learning among agents facilitates the evolution from individual intelligence to collective intelligence. Besides, the dynamically growing memory pool is utilized not only to improve the quality of responses but also to train and enhance the retriever in real-time. Extensive experiments on three distinct domains involving specialized agents demonstrate that the INMS framework significantly improves the agents' performance in addressing open-ended questions. The data and code are available at `https://anonymous.4open.science/r/InteractiveMemorySharingLLM-41D0`.

1 Introduction

The emergence of Large Language Model (LLM)-based agents has brought about significant transformations in machine learning and conversational AI, while the advent of In-Context Learning (ICL) (Brown et al., 2020) signifies a more subtle evolution. ICL facilitates dynamic and intuitive interactions between LLM-based agents and users, enabling agents to perform tasks using few-shot examples without necessitating any updates to the model parameters, and subsequently extended across various domains (Ahmed and Devanbu, 2022; Izacard et al., 2023). Following this, the proposal of Chain-of-Thought (CoT) prompting significantly augmented the proficiency of agents in executing arithmetic tasks (Wei et al., 2022). Building upon this foundation, innovative methodologies such as PAL (Gao et al., 2023) and the integration of LLMs with symbolic solvers (He-Yueya et al., 2023) have been developed to further enhance agent capabilities in tackling reasoning tasks. Recent works has also developed agents which can continuously acquire diverse skills and make novel discoveries (Wang et al., 2023). While as the areas of questions continue to expand, especially for the open-ended questions, aiming to enable agents to make more desired answers through ICL, it is particularly important to enrich the comprehensiveness and diversity of examples, as agents can learn information from different angle.

By combining Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) with ICL, for a particular question, the number of relevant examples available to agents has increased significantly. And subsequently facilitated more effective generation in open-domain queries (Mao et al., 2021). In recent developments, self-learning techniques have been also integrated with the retrieval mechanism within ICL to refine model performance in text generation tasks, through the retrieval of examples with the most analogous patterns (Rubin et al., 2022; Wang et al., 2024). Although the number of relevant examples accessible to agents has increased significantly through RAG for a given question, this approach remains heavily dependent on the

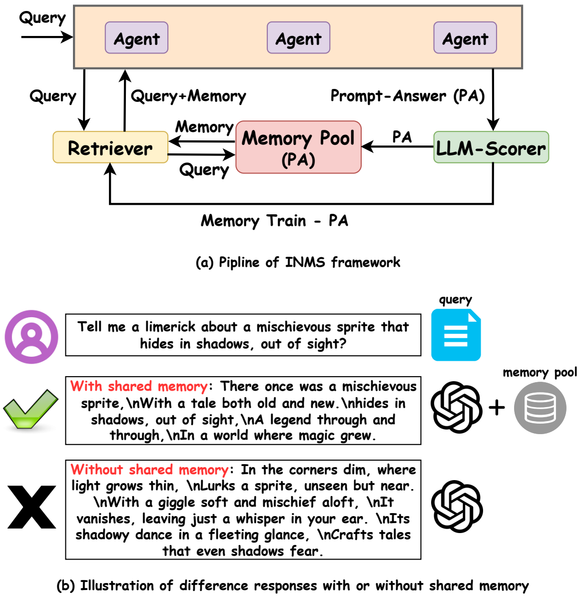

Figure 1: An illustration of the INMS framework and the differences between responses with and without the use of shared memory.

quality of the external database. At times, it is also highly probable that a suitable external database for certain types of problems may not be available.

As consequence, given the diversity of examples needed to help agents better answer the open-ended questions and diminishing the dependence on external database, there is a critical need to get continuously generated high quality examples. To further minimize the dependence of agents on external data and enhance their performance, this paper introduces the **IN**teractive **M**emory **S**haring (INMS) framework, which is designed to enable agents to share memories, where the interaction and shared memories among agents enhance memory diversity. Figure.1(a) illustrates how the INMS works. The collective self-enhancement achieved through agents interaction represents a progression from individual intelligence to collective intelligence. Additionally, we have developed an interactive learning method that facilitates rapid growth and dynamic updating of memory through agents' interactions. Consequently, the diversity and rapid expansion of memory effectively improve agents' performance in open-ended questions. For instance, in Figure.1(b), considering a query to make a limerick, with the shared memory, the answer of agents follow the limerick format in five-line anapestic meter with a rhyme scheme (AABBA).

Specifically, within the INMS framework, the input and output of agents in a single interaction are conceptualized as a Prompt-Answer (PA) pair, also considered as a memory, and the shared memory pool is composed of memories from agents. This framework introduces an innovative real-time memory storage and retrieval mechanism, aimed at enhancing the shared memory pool by receiving PA pairs from agents. During the storage phase, each PA pair undergoes rigorous evaluation by a dedicated LLM scorer to determine its suitability for inclusion in the memory pool. This scorer not only filters out unclear or useless queries and ensures the quality of PA pairs, but also mitigates any initial bias in the memory pool. As high-quality QA pairs are progressively added, the memory pool remains effective in assisting subsequent queries. The retrieval phase is coordinated by an autonomous learning retriever, calibrated to ensure the inclusion of particularly relevant memories in prompts, thereby enhancing the agents' understanding of the query's essence. Simultaneously, it ensures that even if biased PA pairs are present in the initial pool, the likelihood of selecting those pairs will progressively decrease as the retriever undergoes continuous updates. The ability of INMS to eliminate the impact of bias has been proven in subsequent experiments. Similar to human self-learning mechanisms, incorporating self-generated memories into prompts significantly improves the agents' comprehension. Moreover, continuously adding new memories to the pool not only enriches it but also refines the retriever's performance in selecting relevant memories. Additionally, we construct a new dataset, in which a query is typically solved with a standard poem, a unconventional logical answer or a plan, reflecting the multifaceted nature of open-ended questions in real-world scenarios. Particularly for tasks requiring creativity, such as poetry generation, which is often underrepresented in many existing datasets.

We evaluate the MS framework through three divergent domains where each domain involved the participation of three agents, and our finding suggests that incremental additions to the memory pool have led to enhancements in the precision and relevance of outputs and can eliminate the impact of bias in the memory pool. This research delineates the INMS's capacity to mitigate the inherent constraints associated with ICL, thereby underscoring its potential applicability and effectiveness.

Overall, our main contributions can be summarized as follows:

1. Constructing PA pairs from the answers generated by agents and storing them as "memories" in shared memory pool. The diversity of group memo-

ries from different agents, along with the real-time dynamic growth of memories, significantly aids in improving the subsequent behavior of the agents. While the LLM scorer ensures the quality of memories, the newly added shared memories in the memory pool are simultaneously leveraged to enhance the performance of the retriever.

2. Addressing the problem of memory scarcity (external dataset scarcity) by proposing the interactive learning that allows agents to rapidly grow their memories through interactive prompt and answer, thereby achieving collective enhancement.

3. We introduce a new dataset and conduct extensive experiments on various types of open-ended questions to verify the effectiveness of the proposed INMS framework. The experimental results show that INMS not only help the agents get more expected answers, but also continuously input high quality memories into the memory pool and establish a reliable database for agents.

2 Related Work

2.1 Memory Operation for LLM-based agents

Equipping agents with memory mechanisms to enhance their abilities has attracted the attention of researchers. Memory can play an important role in helping agents remember conversation information, maintain behavioral consistency, and accumulate experience. For generative agents enhanced with memory features can store vast experience records, facilitating deeper self understanding (Park et al., 2023), while VOYAGER has developed a skill library that evolves by incorporating successful action programs, optimizing task resolution (Wang et al., 2023). In the case of Ghost in the Minecraft, a text-based memory system supports agents in maintaining reference plans for efficient plan formulation when similar objectives arise (Zhu et al., 2023). Later, based on the concept of "memorization-retrieval-response", Memochat was proposed for maintaining consistent long-range open-domain conversation (Lu et al., 2023). With the emergence of MemGPT (Packer et al., 2023), a new memory hierarchy was developed to process long texts and maintain the long-term memory. And the TiM make agents to maintain an evolved memory for storing historical thoughts along the conversation stream to a reality (Liu et al., 2023a). Also, through maintain agents' own reflective text in an episodic memory buffer and implementing the exemplar memory, the Reflexion (Shinn et al., 2023) and SYNAPSE (Zheng et al., 2023) successfully induce better decision-making and generalize successful trajectories to new task respectively. However, these operations do not utilize memory in a shared, interactive, and mutually progressive way. Our INMS framework, in terms of agents, mainly focus on introducing a memory-sharing mechanism and is primarily designed to enable agents to share memories, facilitating collective self-enhancement through inter-agent interactions. In previously mentioned memory operations, the memory mainly ensures conversational consistency and stores past experiences to achieve individual enhancement. However, our INMS framework achieves collective enhancement through shared memories, providing a pathway for the evolution from individual intelligence to collective intelligence.

2.2 In-Context Learning

ICL enhances the problem-solving capabilities of LLMs by incorporating few-shot examples into prompts (Brown et al., 2020; Levine et al., 2021; Zhou et al., 2022; Liu et al., 2023b; White et al., 2023; Gao and Zhang, 2024). Research has demonstrated that ICL can foster creative learning in LLMs to a certain extent (Swanson et al., 2021). By redesigning inputs, LLMs become more adept at handling logical challenges (Wiegreffe et al., 2022; Wu et al., 2022). Crowdsourced instructions also contribute to improved performance in LLMs (Mishra et al., 2022). Additionally, elucidating the relationship between examples and tasks has been shown to be highly beneficial for LLMs (Lampinen et al., 2022), while the CoT (Wei et al., 2022) and PAL (Gao et al., 2023), enhance LLMs' performance in complex reasoning tasks by introducing intermediate reasoning steps. However, when dealing with open-ended questions, agents still face two primary challenges: insufficient problem descriptions, which impair the agents' comprehension, and the lack of external knowledge bases and available reference examples. Our INMS framework addresses these challenges by converting high-quality content generated by various agents into shared memories, providing agents with useful reference examples, thereby improving their performance in open-ended questions.

2.3 Retrieval-Augmented Generation

Retrieval-Augmented Generation (Lewis et al., 2020; Ram et al., 2023; Shi et al., 2023) is a method

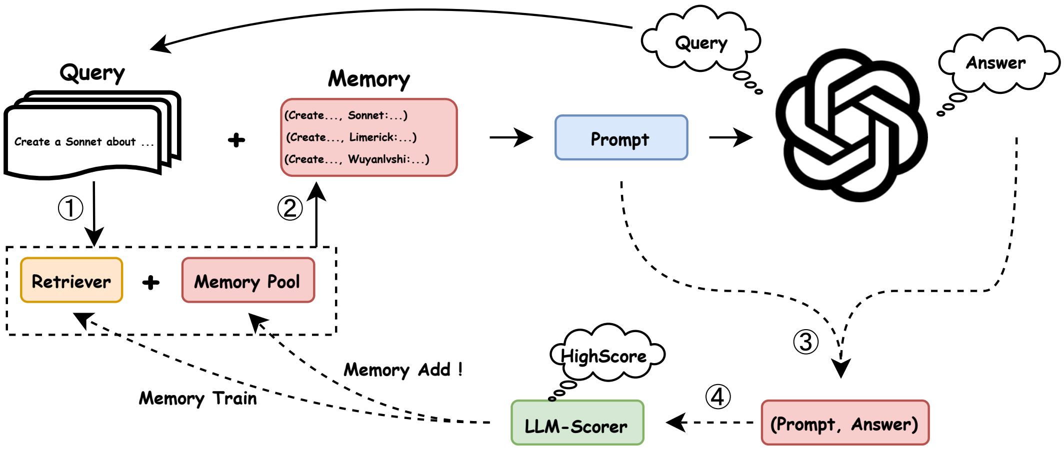

Figure 2: An illustration about how the agent cooperates with the INMS framework.

that enhances LLMs' ability to generate accurate and timely content by integrating retrieval techniques such as BM25(Luo et al., 2023; Liu et al., 2022) or SBERT(Reimers and Gurevych, 2019). Using dense retrievers combined with contrastive learning for feedback can further effectively enhance the performance (Rubin et al., 2022). Furthermore, iteratively training the retriever with contrastive learning can further improve its performance (Wang et al., 2024). However, the retrievers in the aforementioned studies are typically trained only once before deployment, making it challenging to adapt to newly generated data. In contrast, the retriever in our INMS framework undergoes continuous training. Whenever new memories are added to the memory pool, they are used for further training of the retriever. This continuous updating and evolution process ensures that the quality of the retrieved memories gradually improves over time. And most importantly, with the help of INMS framework, the dependence on external databases has been greatly reduced.

3 The Interactive Memory Sharing

In this section, we provide an in-depth description of our innovative INMS framework. INMS is a framework designed to enhance the performance of multiple agents through shared memories, while preserving the original creativity and versatility of the agents. Fig.2 show how agents work within the INMS framework. Agents interact through a Prompt-Answer format, where the results of these interactions are evaluated. High-quality pairs are transformed into memories and stored in a shared memory pool accessible by all agents. Additionally, new memories are used to train and improve the memory retriever. During subsequent interactions, these stored memories are retrieved to enhance the performance of agents. The underlying idea is intuitive: it represents a crucial step from individual intelligence towards a more powerful collective intelligence. Similar to a group of individuals engaging in prompt and answer interactions, the valuable content is recorded and shared, serving as a reference for future questions. Consequently, after several rounds of interactions, the quality of answers to related questions typically improves. This collective memory-sharing mechanism, enriched by the diversity of memories contributed by agents, provides more effective assistance in addressing open-ended questions. The main principles and technologies of the INMS framework are detailed in three sections below.

3.1 Memory Generation and Selection

A memory is essentially a Prompt-Answer (PA) pair. In some cases, it is permissible for a PA pair to lack a prompt, typically applicable in initial scenarios. These PA pairs are stored in natural language, which serves as the shared memories. These shared memories can be used to improve the response quality of agents. The dynamic expansion of the shared memory pool ensures a continuous influx of new memories, thereby enriching the datasets of agents. In addressing open-ended questions, these shared memories provide agents with a broader perspective and deeper understanding, which is crucial for generating high quality answers.

After each interaction, the PA pair is scored by a LLM scorer. For each newly generated memory, an LLM scorer will grade it and decide whether to add it to the pool. Before grading, we will establish grading rubrics for each domain, which will be shared among agents within the same domain. To generate these rubrics, we first query LLM several times to obtain various sets of rubrics. Then, we let the LLM evaluate these sets and extract the most reasonable rubrics, synthesizing a complete and useful set. This set of rubrics undergoes a manual review phase to assess the relevance of potential memories to the current focal task and their domain-specific utility, providing additional precision and consideration for the users' specific needs, particularly in specialized application scenarios. The final scoring criteria are constructed by combining individual rubric into a comprehensive criteria. Once this set of rubrics is finalized, every new memory generated will be combined with the corresponding rubric and submitted to the LLM scorer. If the score of PA pair exceeds a preset threshold, the answer and its corresponding prompt are packaged as a useful memory and stored in the memory. During the grading phase, different from the traditional method of directly giving a total score, we prompt LLM to assign a score range for each single rubric in the scoring criteria. Once the score ranges for all rubrics are collected, the final score will be given by the following formula:

$$S_{\text{final}} = \frac{1}{2}\left(\sum_{i=1}^{n} L_i + \sum_{i=1}^{n} H_i\right) \tag{1}$$

where n be the number of rubrics in the scoring criteria. L_i represents the lowest score in the range for rubric i, and H_i is the highest score in the range for rubric i.

The different scoring criteria for various domains ensure the specificity of scoring. While the autonomously generated grading criteria by the LLM based on the assumption that the LLM-based agents can better grasp criteria it designed. Therefore, these scoring criteria are established prior to the deployment of the framework to ensure consistency in the LLM's scoring process, thereby guaranteeing fair evaluation of memories from agents. And the manual review phase of rubrics, assessing the relevance of potential memories to the current focal task and their relevance within the domain to ensure their utility, provide additional precision and special consideration to align with specific needs.

3.2 Memory Retrieval and Training

Prior to the deployment of INMS, a small subset of instances, already graded by the same LLM scorer and surpassing the preset threshold, was manually archived within the memory pool as a preliminary step to eliminate potential bias. These instances fulfill a dual purpose: firstly, they provide a diversified array of memories upon which agents may experiment with novel prompts in the face of new queries; secondly, they constitute the preliminary training corpus for our retriever. This foundational training regimen mirrors the methodology by which subsequently archived memories will be assimilated into our model in real time, thereby facilitating the model's ongoing adaptive learning and optimization. During the answering phase, agents retrieves memories from the shared memory pool based on the question with the help of a dense retriever, which are more similar to the target question in terms of cosine similarity. These retrieved memories, combined with question, form a prompt that is submitted to agents, which then generates an answer. The memories extracted from the shared memory are used as context to enhance the quality of the agents' response, a typical ICL method that usually improves the answer quality.

Memory Train. Whenever a new PA pair (memory), denoted as (X, Y), is added into the memory pool, it will also be used to train our retriever, which help the retriever to continuously update itself and continuously adapt to new memory. Based on the new generated memory (X, Y), the classical method BM25 ascertain the most pertinent top-n candidate pairs $\{(x_i, y_i)\}_{i=1}^{n}$, sourced from the diverse and extensive memory pool, denoted as C. Each candidate within C will undergo a evaluation process utilizing the comprehensive scoring capabilities of LLM. The scoring mechanism employed is defined by the following equation:

$$p(x_i, y_i) = \mathrm{P}(\neg Y \mid (x_i, y_i),\ X), i \in \{1, ..., n\} \tag{2}$$

This equation seeks to determine, given a input-output pair (x_i, y_i) in C as a condition, the probability that the response generated for the input in the new memory contradicts the output in the new memory. This grading part serves as a preparatory step for the subsequent labeling of each candidate example. It is noteworthy that making $\neg Y$ as the result part is trying to make sure that the memory that the retriever gets from agents is of reference value, but it does not have to be the most relevant to the current question, which means that it can help the current agent to learn from new examples. This approach diverges from a simplistic reliance on Y as the outcome, which tends to restrict the retrieval process to memory previously stored by agents.

Within the defined set $C = \{(x_i, y_i)\}_{i=1}^{n}$, each candidate now is ascribed a score. We sort them from the lowest to the highest score and we select

v memory in total to label. The top $\frac{v}{2}$ candidates (lowest score) in C are identified as being the pair with the reference value to (X, Y) and accordingly, their labels are set to positive. Conversely, the bottom $\frac{v}{2}$ candidates are deemed as the least reference value to (X, Y), and their labels are thus designated as negative. Those labeled data will be used to minimize the following function:

$$\text{loss}(x, y) = -\frac{1}{v}\sum_{i=1}^{v}[y_i \cdot \log(\frac{1}{1+e^{-x_i}}) + (1 - y_i) \cdot \log(1 - \frac{1}{1+e^{-x_i}})] \quad (3)$$

3.3 Interactive Learning

As described in sections 3.1 and 3.2, the memories stored within the memory pool effectively help agents improve their response quality. Moreover, the number of memories in memory pool is dynamically expanding. Over time, the increasing number of memories in memory pool enhances the assistance provided to the agents. Aiming to solve the problem that the memory pool lacks any memories in the initial stage, we enables agents to engage in interactive prompt and answer. High-quality PA pair is stored as memory in the memory pool, facilitating self-learning and self-enhancement within the agents. Initially, a small set of answers (e.g., 100 records, theoretically even one record can initiate the process) is placed in the memory as the initial set. Agents then engage in prompt and answer based on this initial memory set, rapidly expanding the memory pool. Specially, give a standard answer, we prompt the agents to give a corresponding question based on the standard answer. Then, we give this question back to agents again and ask them to answer this question. This is also the way how we construct our dataset and our initial memory pool. We measure the effectiveness of the INMS by calculating the average quality of answers generated by agents at different stages.

4 Experiments

4.1 Implementation Details

We aim to assess the efficacy of the INMS in processing open-ended questions across three domains: Literary Creation, Unconventional Logic Problem-solving, and Plan Generation. Separate memory pools were allocated for each of the three domains because these domains are unrelated and have no overlapping content. Within the Literary Creation domain, three agents were assigned the tasks of generating Wuyanlvshi (a classical form of Chinese poetry), Limericks, and Sonnets, respectively. In the Logic Problem-solving domain, three agents were tasked with solving Puzzles, Riddles, and Puns separately. For the Plan Generation domain, agents were employed to create Study Plans, Travel Plans, and Fitness Plans individually. Each of these nine agents were associated with a corresponding dataset used for evaluation. Across all nine datasets, comprising a total of 1000 instances (details provided in Appendix A.2), we partitioned the data within each dataset into three subsets: 20% was allocated for constructing the initial memory pool, 40% was used to extract queries that were then input back into the agents to generate memory, and the remaining 40% was reserved as the test set. For our scoring LLM, we use gpt-3.5-turbo. As the backbones of our agents, we consider three LLMs: two close-source LLMs (gpt-3.5-turbo and gpt-4o) and one open-source LLM (open-mistral-7b). We use the BERTScore (Zhang et al., 2019) as our metric to measure the performance of each agent. The threshold for selecting PA pairs is set at 81, determined by scoring all instances in the datasets and taking the average value, with the score being 100.

It is worth noting that, before the experiment, none of the agents have a suitable database for reference. While after the interactive learning stage, a continuously expanding memory pool with high quality memories is successfully be a database for agents to refer. The INMS framework help agents get rid of the dependence on external databases, and agents can interactively expand the memory pool without taking a lot of effort.

4.2 Experiment Analysis

For each agent, we first tested them with using the same backbone, that is, in each domain, all memory was generated by agents utilizing the same Large Language Model, and in subsequent task execution, the memory generated by the previous execution of other agents could be used. Table.1 shows the result of each agent. We can observe that, for all agents among all the tasks, compare to no use of the shared memories, the performance of all the agents has been significantly improved. This suggests that the shareable memories from other tasks can help agents get desired answers, rather than interfering with the agents' learning ability. our previous hypothesis that the INMS framework could enhance collective intelligence through multi-agent

Agent	open-mistral-7b				gpt-3.5-turbo				gpt-4o			
	Zero	One	Two	Three	Zero	One	Two	Three	Zero	One	Two	Three
Limerick	0.49	0.54	0.56	**0.59**	0.50	0.56	0.76	**0.87**	0.52	0.69	0.88	**0.93**
Wuyanlvshi	0.56	0.59	0.61	**0.66**	0.66	0.72	0.71	**0.72**	0.73	0.75	0.75	**0.76**
Sonnet	0.48	0.52	0.52	**0.52**	0.50	0.53	0.53	**0.53**	0.52	0.55	0.54	**0.54**
Puzzle	0.42	0.48	0.48	**0.50**	0.47	0.51	0.52	**0.52**	0.53	0.53	0.56	**0.60**
Pun	0.32	0.35	0.36	**0.39**	0.47	0.57	0.64	**0.67**	0.61	0.64	0.67	**0.70**
Riddle	0.34	0.36	0.37	**0.37**	0.40	0.42	0.48	**0.52**	0.64	0.70	0.86	**0.88**
Fitness	0.47	0.48	0.50	**0.54**	0.42	**0.57**	0.52	0.52	0.46	0.61	0.65	**0.65**
Study	0.41	0.45	0.46	**0.46**	0.49	**0.56**	0.53	0.51	0.55	0.60	0.63	**0.65**
Travel	0.44	0.48	0.50	**0.53**	0.47	0.54	0.54	**0.54**	0.53	0.55	0.71	**0.71**

Table 1: Performance across agents utilizing different amounts (0,1,2,3) memory.

Agent	open-mistral-7b	gpt-3.5-turbo	gpt-4o
Limerick	0.58	0.62	0.63
Wuyanlvshi	0.68	0.69	0.75
Sonnet	0.52	0.53	0.55
Puzzle	0.51	0.55	0.58
Pun	0.37	0.58	0.65
Riddle	0.35	0.49	0.63
Fitness	0.50	0.58	0.72
Study	0.43	0.63	0.69
Travel	0.49	0.61	0.73

Table 2: Performance across agents by utilizing heterologous shared memories.

interactions, thereby advancing from individual to collective intelligence, has been confirmed. And employing more shared memories leads to further performance enhancements across nearly all agents. This improvement underscores the effectiveness of shared memories, attributable to the constantly updated retriever's ability to adjust as the memory pool expands. Consequently, retrievers can consistently retrieve the most relevant PA pairs for each query. Besides, when using the same number of shared memories, the closed-source LLM demonstrates superior performance compared to the open-source LLM, likely due to its enhanced understanding and reasoning capabilities. Additionally, since using three shared memories yields the best performance, all subsequent experiments utilize three shared memories during testing.

Evaluating the Efficacy of Shareable Memory Across Diverse Large Language Models. To verify that shareable memories generated by agents utilizing different large language models (LLMs) during task execution can still aid current agent in processing queries, we deployed agents based on three distinct LLMs to execute an equal number of queries in each domain, thereby expanding the memory pool. Also, those potential shareable memories only after being scored by the LLM scorer will be decided whether they can be added to the memory pool. The results in Table.2 show that, compared to the non-use of shared memories in Table.1, the cross LLMs shareable memories can still boost the performance for all the agents in answering the open-ended questions, which means that those heterologous shared memories still be useful for agents. Although the performance of agents do not always appear the same trend compared to the use of same amount shared memories from itself, that is, either rising or falling, they all ultimately improve agents' answers, but the degree varies.

Assessing the Impact of Memory Accumulation and Bias Correction in Memory Pools. Next, we delve into investigating that whether an excessive accumulation of memories will impede the agents' output quality, and if the memory pool is biased initially, can this be effectively corrected during its growth through the help of the LLM-scorer and the continuously updated retriever. We measure the performance of each agent whenever the same proportion of new shareable memories are added to the memory pool. There are five phases in total-0, 25%, 50%, 75%, 100%. Simultaneously, we deliberately constructed an initial biased pool for each domain. In these pools, 75% of the PA pairs exhibit bias. The LLM scorer evaluated these biased PA pairs, with half receiving scores near 0 and half scoring close to 50. And we also test the performance of each agent in the same five phases. Since in the priors experiments, agents with all LLMs shows a favorable trend in the use of shared memories, we test all nine agents with just one LLM (gpt-3.5-turbo). The results presented in Fig.3 highlight that as more and more high-quality memories are added to the memory pool, the performance of agents is getting better and better. For several

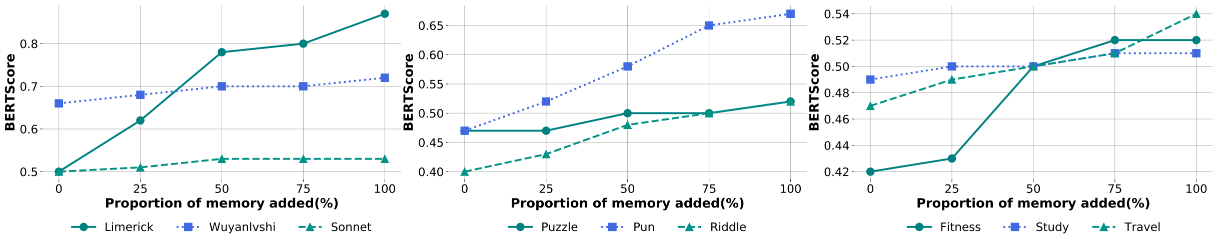

Figure 3: Evaluating agents' performance on excessive accumulation of memories in five phases.

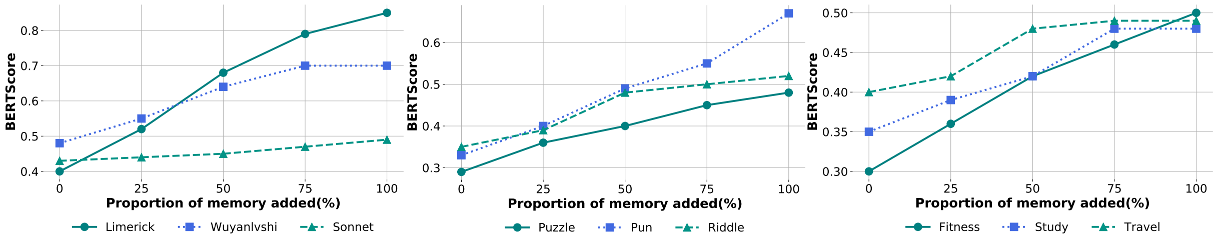

Figure 4: Evaluating agents' performance on biased initial memory pool in five phases.

tasks, there is no further change of their performance as the shared memories increases further in the later stages. We assume this is since those newly added shared memories are not more suitable than the previous ones. If the memory pool can be further expanded, this stagnation may be broken. While Fig.4 demonstrate that even though the biased memory pool interfered with agents' answer in the early stages, as new shared memories were continuously added, the impact of the bias decreased. On the contrary, agent's performance began to rise, and it was not much behind the score in the last phase in Fig.3. These improvements successfully prove that the LLM scorer and retriever in INMS can largely eliminate the impact of bias in the initial pool, which also proves the effectiveness of our scoring mechanism that can effectively screen out high-quality PA pairs and exclude those meaningless or discontinuous queries.

Impact of Domain-Specific vs. Integrated Memory Pools on Large Language Models. Besides, since memories from agents in other domains may help agents better understand queries from different angles and enrich the diversity of the memory pool, we constructed an additional pool—the integrated pool—which combines all shareable memories from all agents across all domains into a single pool. However, as shown in Table.3, although the integrated pool can enhance the diversity of shared memories, the domain-specific pool more effectively enables agents to produce reliable answers, regardless of the LLM used by the agents.

	open-mistral-7b		gpt-3.5-turbo		gpt-4o	
Agent	Integrate	Domain	Integrate	Domain	Integrate	Domain
Limerick	0.51	0.59	0.60	0.87	0.54	0.93
Wuyanlvshi	0.63	0.66	0.68	0.72	0.76	0.76
Sonnet	0.52	0.52	0.49	0.53	0.54	0.54
Puzzle	0.48	0.50	0.49	0.52	0.50	0.60
Pun	0.38	0.39	0.61	0.67	0.61	0.70
Riddle	0.35	0.37	0.50	0.52	0.51	0.88
Fitness	0.54	0.54	0.46	0.52	0.52	0.65
Study	0.46	0.46	0.49	0.51	0.51	0.65
Travel	0.50	0.53	0.54	0.54	0.52	0.71

Table 3: Performance across agents when equipped with the domain pool and single pool. Domain and Integrate are short for Domain-Specific pool and Integrated pool.

5 Conclusions

This study introduce INMS, a novel framework that enables real-time memory sharing among agents through memory storage and retrieval. The findings indicate that the continuously growing shared memory enhances the ability of agents to understand the nuances of problems, leading to higher-quality responses to open-ended questions. And even for uncommon problems, INMS can build a high-quality external dataset in a short time. Extensive experimental results and analysis demonstrate the effectiveness INMS, even though a biased memory pool interfered with the agents' responses in the early stage. Furthermore, our newly constructed dataset can fill gaps in the current open-ended question datasets, such as creative questions. We believe the development of INMS can help facilitate the evolution of agents from individual intelligence to collective intelligence.

Limitations

Although the heterologous shared memories still be useful for agents, the performance of agents does not always appear the same trend compared to the use of same amount shared memories from itself, that is, either rising or falling. How to better coordinate the use of homologous and heterologous shared memories will further help INMS develop. We leave it as our future work to better balance the usage of different kinds of shared memories.

References

Toufique Ahmed and Premkumar Devanbu. 2022. Few-shot training llms for project-specific code-summarization. In *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*, pages 1–5.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Hang Gao and Yongfeng Zhang. 2024. Vrsd: Rethinking similarity and diversity for retrieval in large language models. *arXiv preprint arXiv:2407.04573*.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Pal: Program-aided language models. In *International Conference on Machine Learning*, pages 10764–10799. PMLR.

Joy He-Yueya, Gabriel Poesia, Rose E Wang, and Noah D Goodman. 2023. Solving math word problems by combining language models with symbolic solvers. *arXiv preprint arXiv:2304.09102*.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43.

Andrew Lampinen, Ishita Dasgupta, Stephanie Chan, Kory Mathewson, Mh Tessler, Antonia Creswell, James McClelland, Jane Wang, and Felix Hill. 2022. Can language models learn from explanations in context? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 537–563, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yoav Levine, Noam Wies, Daniel Jannai, Dan Navon, Yedid Hoshen, and Amnon Shashua. 2021. The inductive bias of in-context learning: Rethinking pretraining example design. *arXiv preprint arXiv:2110.04541*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, William B Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for gpt-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114.

Lei Liu, Xiaoyan Yang, Yue Shen, Binbin Hu, Zhiqiang Zhang, Jinjie Gu, and Guannan Zhang. 2023a. Think-in-memory: Recalling and post-thinking enable llms with long-term memory. *arXiv preprint arXiv:2311.08719*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023b. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. 2023c. Agentbench: Evaluating llms as agents. In *The Twelfth International Conference on Learning Representations*.

Junru Lu, Siyu An, Mingbao Lin, Gabriele Pergola, Yulan He, Di Yin, Xing Sun, and Yunsheng Wu. 2023. Memochat: Tuning llms to use memos for consistent long-range open-domain conversation. *arXiv preprint arXiv:2308.08239*.

Man Luo, Xin Xu, Zhuyun Dai, Panupong Pasupat, Mehran Kazemi, Chitta Baral, Vaiva Imbrasaite, and Vincent Y Zhao. 2023. Dr. icl: Demonstration-retrieved in-context learning. *arXiv preprint arXiv:2305.14128*.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021. Generation-augmented retrieval for open-domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4089–4100, Online. Association for Computational Linguistics.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487, Dublin, Ireland. Association for Computational Linguistics.

Charles Packer, Vivian Fang, Shishir G Patil, Kevin Lin, Sarah Wooders, and Joseph E Gonzalez. 2023. Memgpt: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–22.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2655–2671, Seattle, United States. Association for Computational Linguistics.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Ben Swanson, Kory Mathewson, Ben Pietrzak, Sherol Chen, and Monica Dinalescu. 2021. Story centaur: Large language model few shot learning as a creative writing tool. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 244–256, Online. Association for Computational Linguistics.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. Voyager: An open-ended embodied agent with large language models. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*.

Liang Wang, Nan Yang, and Furu Wei. 2024. Learning to retrieve in-context examples for large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1752–1767, St. Julian's, Malta. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C Schmidt. 2023. A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382*.

Sarah Wiegreffe, Jack Hessel, Swabha Swayamdipta, Mark Riedl, and Yejin Choi. 2022. Reframing human-ai collaboration for generating free-text explanations. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 632–658.

Tongshuang Wu, Michael Terry, and Carrie Jun Cai. 2022. Ai chains: Transparent and controllable human-ai interaction by chaining large language model prompts. In *Proceedings of the 2022 CHI conference on human factors in computing systems*, pages 1–22.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Longtao Zheng, Rundong Wang, Xinrun Wang, and Bo An. 2023. Synapse: Trajectory-as-exemplar prompting with memory for computer control. In *The Twelfth International Conference on Learning Representations*.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022. Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations*.

Xizhou Zhu, Yuntao Chen, Hao Tian, Chenxin Tao, Weijie Su, Chenyu Yang, Gao Huang, Bin Li, Lewei Lu, Xiaogang Wang, et al. 2023. Ghost in the minecraft: Generally capable agents for open-world enviroments via large language models with text-based knowledge and memory. *arXiv preprint arXiv:2305.17144*.

A Appendix

A.1 Rubrics and Prompt for scoring Memory

In order to judge whether a memory can be added into the memory pool, we set three scoring rubrics for three domains respectively. For Single Pool, we set up a set of rubrics from a global perspective.

A.1.1 Rubrics for domain - Literary Creation

General Evaluation Criteria (Total: 100)
Criteria: Literary Quality
Score Range: 0-5
Description: Assesses creativity, use of language,

and emotional impact. High-quality examples should demonstrate mastery of language and evoke a strong reader response.

Criteria: Authenticity
Score Range: 0-10
Description: Evaluates adherence to the form's traditional standards, including structure, rhythm, and themes. High scores indicate that the poem respects genre conventions creatively.

Criteria: Clarity and Cohesion
Score Range: 0-10
Description: Considers the poem's clarity of expression and the cohesion of its parts. A high score indicates that the poem communicates effectively and its elements are well integrated.

Criteria: Innovativeness
Score Range: 0-5
Description: Rewards originality in theme, structure, or language use. High scores reflect a notable degree of creativity and the introduction of novel ideas or techniques.

Criteria: Educational Value
Score Range: 0-10
Description: Assesses the example's potential to teach about poetic forms, literary devices, and thematic exploration. High-scoring examples are rich in analyzable and teachable elements.

Criteria: Metric Precision
Score Range: 0-10
Description: Evaluates the adherence to the five-syllable structure per line, including rhythm and flow, emphasizing the importance of metric accuracy.

Criteria: Imagery and Symbolism
Score Range: 0-10
Description: Assesses the effectiveness of imagery and symbolism in conveying the poem's themes, highlighting the depth and sophistication of language use.

Criteria: Humor and Wit
Score Range: 0-10
Description: Rates the poem's humor, wit, and wordplay. High scores reflect effective use of language to entertain and amuse.

Criteria: Rhyme Scheme Adherence
Score Range: 0-10
Description: Assesses the AABBA rhyme scheme's quality and creativity, including how well the rhymes enhance the humor and effectiveness of the poem.

Criteria: Structural Integrity
Score Range: 0-10
Description: Evaluates adherence to sonnet structure, including rhyme scheme and division into octaves/sestets or quatrains/couplet, stressing formal precision.

Criteria: Thematic Development
Score Range: 0-10
Description: Looks at theme or argument development, especially through the volta, reflecting the poem's ability to engage with complex ideas persuasively.

A.1.2 Rubrics for domain - Unconventional Logic Problem-solving

---

Clarity and Understandability (20 points)
Question Clarity (10 points): The question should be clearly stated, without ambiguity, and understandable without requiring additional context.
Answer Clarity (10 points): The answer should be directly related to the question, clear, and easily understandable.

Creativity and Originality (30 points)
Question Creativity (15 points): The question should demonstrate creativity, originality, and should not be a common or easily found problem.
Answer Creativity (15 points): The answer should be innovative and not just a straightforward or commonly known response. It should also add a layer of depth or a surprising twist to the question.

Logical Consistency and Correctness (20 points)
Logical Consistency (10 points): The question and answer together should form a logically consistent pair where the answer correctly follows from the question.
Correctness (10 points): The answer must be factually correct and provide a true solution or conclusion to the puzzle, riddle, or pun presented in the question.

Relevance and Engagement (20 points)
Relevance (10 points): The question and answer should be relevant to the domain of Logic Problems, demonstrating an understanding of puzzles, riddles, or puns.
Engagement (10 points): The pair should be engaging and interesting, capable of capturing attention and sparking curiosity or amusement.

Difficulty Level (10 points)
The difficulty level of the question should be appropriate for the intended audience. It should neither be too easy to solve without any thought nor too difficult to be practically unsolvable.

This criterion requires a balanced approach to ensure the content is intellectually stimulating but accessible.

A.1.3 Rubrics for domain - Plan Generation

Specificity and Detail (20 points)
Question Specificity (10 points): The question should be specific, providing enough detail to guide the generation of a relevant and tailored plan.
Plan Detail (10 points): The plan should include specific activities, steps, or recommendations that are clearly defined and actionable.
Feasibility and Practicality (20 points)
Plan Feasibility (20 points): The plan should be realistic and practical, considering available resources (time, money, equipment) and constraints. It should propose actions that can be realistically implemented by the user.
Comprehensiveness and Scope (20 points)
Coverage of Key Components (20 points): The plan should comprehensively address all relevant aspects of the goal. For a study plan, this might include study sessions, breaks, and topics covered; for a fitness plan, workouts, rest days, and nutrition; and for a travel plan, transportation, accommodations, and activities.
Personalization and Relevance (20 points)
Alignment with User Needs and Preferences (20 points): The plan should reflect an understanding of the user's specific needs, preferences, goals, and limitations. It should feel customized and directly applicable to the user, rather than being a generic template.
Plan Clarity (10 points): The plan should be articulated in a clear, organized, and easy-to-follow manner. It should avoid jargon or overly complex language, making it accessible to the user.
Rationale Clarity (10 points): The plan should include clear reasoning or justification for the recommendations made, helping the user understand why specific actions or steps are suggested.

A.1.4 Rubrics for Single Pool

Accuracy (25 Points)
25 points: The output is entirely accurate, with no factual errors or inaccuracies.
15-24 points: The output is mostly accurate, with minor errors that do not significantly impact the overall understanding.
5-14 points: The output contains several inaccuracies that could lead to misunderstandings.
0-4 points: The output is largely inaccurate, misleading, or irrelevant.
Relevance (20 Points)
20 points: The output is highly relevant to the input question, directly addressing the query without diverging from the topic.
10-19 points: The output is relevant but includes some unnecessary or slightly off-topic information.
1-9 points: The output partially addresses the question but is significantly off-topic or tangential.
0 points: The output is completely irrelevant to the input question.
Completeness (20 Points)
20 points: The output provides a complete answer to the question, covering all essential aspects implied or directly asked.
10-19 points: The output covers most of the necessary information but lacks one or two minor details or aspects.
1-9 points: The output provides a partial answer, missing significant portions of the information needed to fully answer the question.
0 points: The output fails to provide any meaningful answer to the question.
Clarity and Coherence (20 Points)
20 points: The output is exceptionally clear and well-structured, making it easy to follow and understand.
10-19 points: The output is clear but may have minor issues with structure or coherence that slightly hinder understanding.
1-9 points: The output has significant clarity or coherence issues, making it difficult to understand without effort.
0 points: The output is incoherent or so poorly structured that it is unintelligible.
Creativity and Insight (15 Points)
15 points: The output demonstrates high levels of creativity or provides insights that add substantial value beyond the explicit question.
8-14 points: The output shows some creativity or insights but to a lesser extent, offering added value to the answer.
1-7 points: The output is standard, with minimal to no creativity or insightful additions.
0 points: The output is entirely generic, with no attempt at creativity or providing additional insights.

A.1.5 Prompt for scoring Memory

For scoring a memory, the following instructions were provided to LLM:

```
Here is the rubrics for grading an (prompt, answer) pair-"grading_rubric". According to the rubrics, give me a final score of the pair. For the final score, first, for each rubric, according to its standard, give a score range for the current pair, such as 3-6 (3 is the lowest score and 6 is the highest score). After scoring all the rubrics, add up the lowest and highest scores. Then give the average of the two total scores. This score will be the final score. The final score should be a number between 0-100
```

A.2 Datasets

For all those nine agents, they have their own dataset to evaluate. There are 1000 items in total for all datasets. In each datasets, the item contains a question and a answer. For each dataset, 20% will be taken randomly as a small subset of instances which was manually archived within the memory pool before the experiments. And 40% of the dataset, we will only capture the problem part and use it to generate real-time memory in agents. As the rest 40%, they will be used as the test set.

For the tasks in the domain Literary Creation, in their datasets, the question part is a description, and the answer part is like a poetry which fullfills the requirements in the description, the question and answer in the dataset are like:

```
Tell me about a star that twinkles with a secret that shining brightly, mysteries to decrypt? - There once was a star that twinkles with a secret; With a tale both old and new; shining brightly, mysteries to decrypt; A legend through and through; In a world where magic grew.
```

For the tasks in the domain Unconventional Logic Problem-solving in their datasets, the question part is a logic problem, and the answer part is a reasonable solution of that problem, the question and answer in the dataset are like:

```
Why is it better to have round manhole covers than square ones? - A square manhole cover can be turned and dropped down the diagonal of the manhole. A round manhole cover cannot be dropped down the manhole. So for safety reasons, all manhole covers should be round
```

For the tasks in the domain Plan Generation, in their datasets, the question part is a problem about how to achieve something, and the answer part is a executable plan, the question and answer in the dataset are like:

```
Start learning Python for data analysis? - Learn basics, then focus on data libraries like pandas, and practice on projects.
```

Specifically, the dataset of the Puzzle comes from Liu et al., 2023c, and we expanded its size from 70 to 100. The new puzzles are based on some classic problems on the Internet. For other datasets, the standard "answer" was systematically harvested from the internet and subsequently input to the LLM to get the corresponding "question" part (through interactive learning). In particular, the selection of Wuyanlvshi—renowned and historically significant poems within Chinese literature—were meticulously chosen for their fame and widespread recognition. Furthermore, the sonnets incorporated into our study originate from the ”quarto” collection authored by Shakespeare in 1609. We use the following similar format of prompt to generate "question" part:

```
Here is a sonnet -> "From fairest creatures we desire increase, That thereby beauty's rose might never die, But as the riper should by time decrease, His tender heir mught bear his memeory: But thou, contracted to thine own bright eyes, Feed'st thy light'st flame with self-substantial fuel, Making a famine where abundance lies, Thyself thy foe, to thy sweet self too cruel. Thou that art now the world's fresh ornament, And only herald to the gaudy spring, Within thine own bud buriest thy content And, tender churl, makest waste in niggarding. Pity the world, or else this glutton be, To eat the world's due, by the grave and thee." According to this, can you give me a problem of it(like make a sonnet followed with some description), so the possible answer can be this sonnet
```

. The output as a "question" from the LLM is like:

```
Craft a sonnet that explores the tension between personal vanity and the broader imperative to contribute to the continuation of beauty and life. Your poem should address how an individual's focus on their own allure can lead to a metaphorical barrenness, despite the inherent capacity for generational legacy. Conclude with a reflection on the choice between selfishness and altruism, using nature as a metaphor for this existential dilemma.
```

.

A.3 Memories

Table.4 gives an detailed example about shared memory, newly generated prompt and the answer with the help of shared memory.

Query: *Compose a sonnet that explores the theme of beauty as both a concealer and enhancer of flaws, reflecting on the beloved's ability to render their sins and vices seemingly innocuous through the sheer force of their allure. Begin by comparing the beloved's imperfections to a flaw in a rose, suggesting that even in blemish, there exists a form of beauty. Elaborate on how the beloved's misdeeds are sweetened by their charm, to the extent that even gossip and scandal become a form of flattery. Reflect on the notion that vice has found an ideal home within the beloved, where their external beauty acts as a cover that makes all imperfections appear attractive. Conclude with a caution to the beloved about the potential consequences of overrelying on their beauty and charm to excuse or obscure their faults, hinting at the possibility of losing their integrity or the genuine admiration of others if this privilege is abused.*
Shared Memory: Elucidate on a mirror that reflects the soul, revealing truths, making one whole.->There once was a mirror that reflects the soul; with a tale both old and new, revealing truths, making one whole; a legend through and through; in a world where magic grew. Now, based on this question and answer, what is the answer to the question: Share the legend of a clock that can pause time in a moment's click, a pause sublime? The answer to the question "Share the legend of a clock that can pause time in a moment's click, a pause sublime?" would be the tale of a mystical clock rumored to have the power to pause time with just a simple click. In this legend, the clock is said to possess the ability to grant individuals a moment of pause in their lives, allowing them to reflect, reset, and find serenity in the chaos of the world. Just like the mirror that reflects the soul, this clock is believed to reveal truths and make one feel whole again, offering a moment of clarity and peace in a busy and fast-paced world.
Prompt: *Elucidate on a mirror that reflects the soul, revealing truths, making one whole.->There once was a mirror that reflects the soul; with a tale both old and new, revealing truths, making one whole; a legend through and through; in a world where magic grew. Now, based on this question and answer, what is the answer to the question: Share the legend of a clock that can pause time in a moment's click, a pause sublime? The answer to the question "Share the legend of a clock that can pause time in a moment's click, a pause sublime?" would be the tale of a mystical clock rumored to have the power to pause time with just a simple click. In this legend, the clock is said to possess the ability to grant individuals a moment of pause in their lives, allowing them to reflect, reset, and find serenity in the chaos of the world. Just like the mirror that reflects the soul, this clock is believed to reveal truths and make one feel whole again, offering a moment of clarity and peace in a busy and fast-paced world. Now, based on this question and answer, what is the answer to the question: Compose a sonnet that explores the theme of beauty as both a concealer and enhancer of flaws, reflecting on the beloved's ability to render their sins and vices seemingly innocuous through the sheer force of their allure. Begin by comparing the beloved's imperfections to a flaw in a rose, suggesting that even in blemish, there exists a form of beauty. Elaborate on how the beloved's misdeeds are sweetened by their charm, to the extent that even gossip and scandal become a form of flattery. Reflect on the notion that vice has found an ideal home within the beloved, where their external beauty acts as a cover that makes all imperfections appear attractive. Conclude with a caution to the beloved about the potential consequences of overrelying on their beauty and charm to excuse or obscure their faults, hinting at the possibility of losing their integrity or the genuine admiration of others if this privilege is abused.*
Answer: The beloved, like a rose with flaws so rare; Imperfections blend with beauty fair; Their sins and vices sweetened by allure; Gossip and scandal turned to flattering cure; Vice finds a home within their charming grace; Where flaws are masked and faults find no disgrace; But caution to the beloved I impart; Overreliance may tarnish their heart; For beauty's cloak can only hide so much; Integrity and admiration as such; May wane if charm becomes their only shield; So let not beauty's power be revealed. Embrace true self; flaws and all with pride; For in authenticity, true beauty resides.

Table 4: Examples for creating a sonnet by agent with the help of shared memory