AI Town: A Study on Virtual Society Simulation Based on Generative Agents and Retrieval-Augmented Generation (RAG)

ZengKaibo
MC566823
and ShengJunwei
MC566896
and LiJiayi
MC566832

Abstract

This project aims to construct an "AI Town" for simulating trustworthy human social behaviors by reproducing and extending the Generative Agents architecture proposed by Stanford University and Google (Park et al., 2023). This core architecture endows agents with the capabilities of long-term behavioral consistency, memory, reflection, and planning. To address the inherent limitations of Large Language Models (LLMs)—such as lack of domain-specific knowledge and the "hallucination" problem—we integrate Retrieval-Augmented Generation (RAG) technology into the framework. The RAG system is responsible for efficiently storing and retrieving all actions, dialogues, and internal thinking processes (e.g., planning and reflection) of the agents, thereby enabling transparent and traceable querying and analysis of agent behaviors. Through in-depth Prompt Engineering, we assign agents specific "sense of presence" roles and observe the resulting communication and emergent social behavior patterns. Key achievements of the project include the in-depth localization of AI Town into Chinese, optimization of Chinese prompts to interface with Chinese LLMs, and support for full local deployment (accessing LLMs and embedding models via Ollama). This integrated framework provides researchers with a dynamic, controllable platform to explore complex interactions and emergent phenomena in LLM-driven virtual societies.

1 Introduction

1.1 Motivation

Trustworthy human behavior agent models can support interactive applications such as immersive environments, interpersonal communication rehearsal spaces, and prototyping tools (Park et al., 2023). This project aims to reproduce and extend the open-source Generative Agents project developed by Stanford University and Google, constructing an **"AI Town"** to simulate credible human behaviors. By extending Large Language Models (LLMs), Generative Agents can simulate individual behaviors (e.g., waking up, cooking, working) as well as emergent social behaviors (e.g., forming opinions, initiating dialogues, making plans, and reflecting on the past) (Park et al., 2023). However, LLMs face inherent limitations, including **lack of domain-specific knowledge, the "hallucination" problem, and substantial computational resources required for updating internal knowledge**. To address these challenges, Retrieval-Augmented Generation (RAG) has emerged as a solution—it can provide reliable and up-to-date external knowledge to enhance the quality of generated content. This project integrates RAG technology into the AI Town framework, which not only strengthens agents' ability to access historical information but also facilitates efficient querying and analysis of agents' thinking and action histories for researchers. The core research motivation of our project is to assign agents specific **"sense of presence"** or role settings through **Prompt Engineering**, and observe and record the communication and behavioral patterns among agents under this configuration.

1.2 Achievements

This project has successfully accomplished the following core tasks:

1. **Reproduction and In-depth Localization of AI Town into Chinese**: Based on the wounderland project, we reconstructed and implemented **in-depth Chinese localization** of the original Generative Agents. We rewrote all prompts to switch the agents' "native language" to Chinese, and optimized Chinese prompts as well as the dialogue initiation/termination logic between agents to align with the capabilities of Chinese LLMs (e.g.,

Qwen or GLM-4). Additionally, maps and character names were localized into Chinese simultaneously to prevent LLMs from switching to an English context in mixed Chinese-English scenarios.

2. **RAG System Integration**: We developed and integrated a RAG system for searching and retrieving agents' thinking and action history. The introduction of RAG enables more efficient traceability and analysis of agent behaviors.

3. **Support for Local Deployment**: We added support for the local Ollama API and connected LlamaIndex's embedding functionality to Ollama, achieving **full local deployment** and thereby reducing experimental costs. The project also fixed issues in the original version (e.g., agents failing to wake up after falling asleep) and added features such as "checkpoint recovery".

1.3 Literature Review

The research foundation of this project focuses on two key areas: **Generative Agents** and **Retrieval-Augmented Generation (RAG)**.

1.3.1 Generative Agents

Proposed by Stanford University and Google, Generative Agents aim to simulate credible human behaviors (Park et al., 2023). Their core lies in an architecture that extends large language models, comprising three key components:

1. **Memory Stream**: A long-term memory module that records the agent's complete experiences in natural language. The memory retrieval model dynamically extracts records required to guide the agent's immediate behaviors by integrating **relevance**, **recency**, and **importance**.

2. **Reflection**: Over time, this component synthesizes memories into higher-level inferences, enabling agents to draw conclusions about themselves and others to better guide their behaviors. These inferences are themselves fed into the memory stream.

3. **Planning**: This mechanism converts these conclusions and the current environment into high-level action plans, which are recursively refined into detailed behaviors. Planning helps agents maintain behavioral consistency and credibility over extended timeframes.

This architecture is driven by powerful **prompting** capabilities, enabling agents to remember, retrieve, reflect, interact, and plan. Furthermore, through the **Role Prompting** mechanism, background settings for agents' "sense of presence" can be assigned, which constitutes part of the agents' identity description.

1.3.2 Retrieval-Augmented Generation (RAG)

RAG is an advanced AI technology that enhances generative models by integrating information from external data sources (knowledge bases). The main advantages of RAG include **alleviating hallucinations** (LLMs inherently tend to generate factually incorrect content) and **knowledge updating** (ability to leverage the latest information from external authoritative knowledge bases, avoiding reliance solely on the model's outdated internal knowledge). The RAG framework typically consists of three main processes: Retrieval, Augmentation, and Generation (Fan et al., 2024). The retrieval phase usually involves indexing (splitting documents into chunks and encoding them into vectors) and retrieving the most relevant chunks based on semantic similarity. The development of RAG technology has undergone different paradigms, such as **Naive RAG**, **Advanced RAG**, and **Modular RAG** (Gao et al., 2023).

1.4 Conclusion

A key challenge of the Generative Agents architecture is ensuring that the long-term experiences most relevant to the agent's current behavior are retrieved and integrated within the limited context window. RAG technology, particularly its retrieval and augmentation mechanisms, provides a framework-level solution to this challenge. By applying the concept of RAG to agents' **memory retrieval and historical querying**, this project aims to construct an AI society that can both simulate complex social behaviors and provide transparent and traceable behavioral histories. Meanwhile, Prompt Engineering— as described by Schulhoff et al. (2024)—serves as the **fundamental control language** for driving and regulating agents' cognition and actions.

2 Design and Architecture

The core design of this project is based on the Generative Agents architecture (Park et al., 2023), with

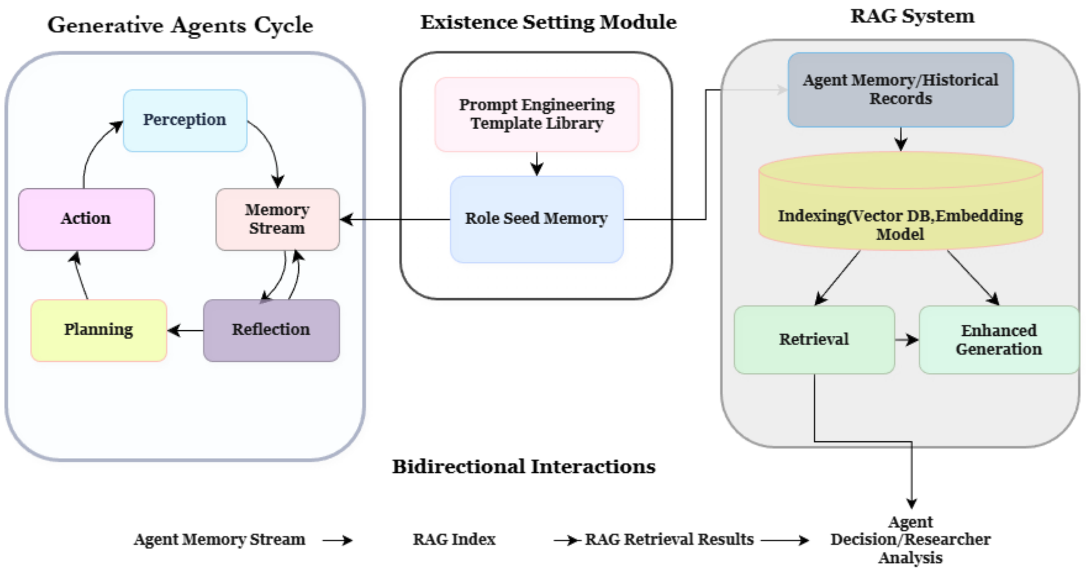


Figure 1: AI Town System Architecture: Synergistic Framework of Generative Agents Cycle, Existence Setting Module, and RAG System with Bidirectional Interactions

the integration of a Retrieval-Augmented Generation (RAG) framework to enhance virtual social behaviors of agents and enable traceable analysis that is shown in Figure 3.

2.1 Customization of Agent Architecture

We adopt the **"Perceive-Remember-Plan-Act cycle"** of Generative Agents. This cycle ensures the coherence and trustworthiness of agents' behaviors.

- **Memory Stream**: At each time step, agents perceive the environment, and all observations are stored in the long-term memory module—the Memory Stream—in natural language. When deciding the next action, the memory retrieval model dynamically extracts required memories by integrating **relevance**, **recency**, and **importance**.

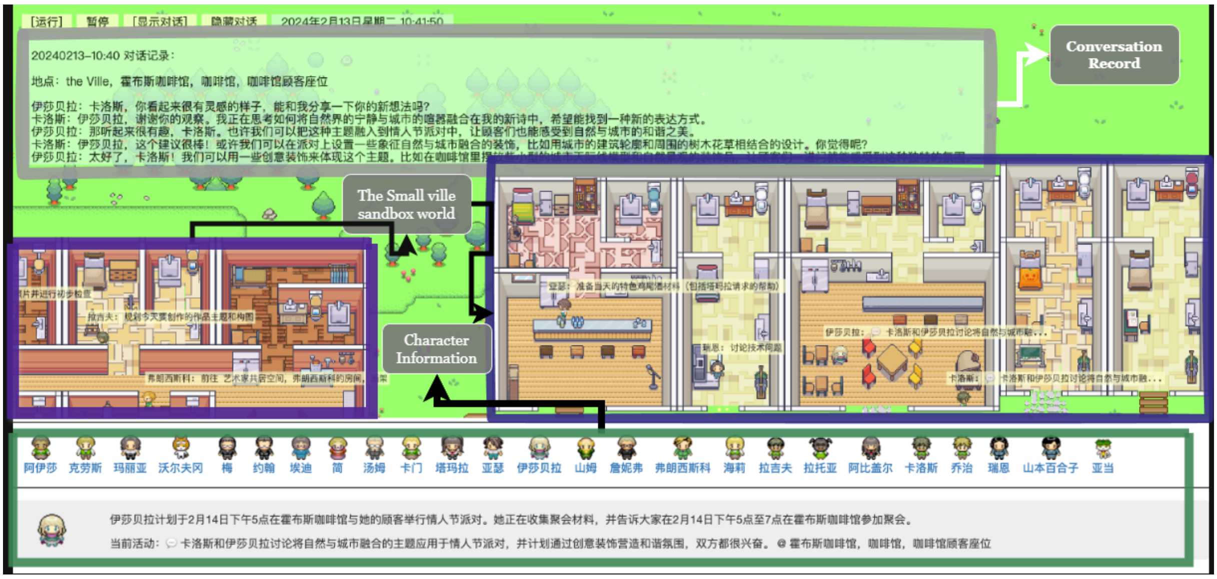
- **Reflection and Planning**: The retrieved memories are used to form long-term **Plans** and generate higher-level **Reflections**. These plans and reflections help agents maintain behavioral consistency and coherence over extended timeframes, and they are themselves fed back into the Memory Stream to influence future behaviors.

- **"Sense of Existence" Setting**: Leveraging the powerful **prompting** capabilities of LLMs, we assign agents specific "sense of existence" roles through **in-depth Prompt Engineering** (Schulhoff et al., 2024). This setting is part of the agents' identity and traits, and is injected as **initial seed memories** at the start of the Memory Stream. This **Role Prompting** mechanism (Schulhoff et al., 2024) guides LLMs to reason and act around this core setting when generating dialogues and behaviors, enabling the observation of specific social behaviors.

2.2 RAG Integration Framework Design

To achieve **transparent and traceable** querying and analysis of agents' behaviors, we developed a RAG system. The core design philosophy of this system is to treat **all actions, dialogues, and internal thinking processes (e.g., planning and reflection)** generated by agents during simulation as external knowledge sources. The workflow of the RAG system follows a typical retrieval-augmented generation structure, consisting of the following key steps:

1. **Indexing**: Each step of the agents' historical records is treated as document chunks. This textual information is **encoded into vector representations** via an embedding model and stored in a vector database. In implementation, the project supports connecting LlamaIndex's embedding functionality to local Ollama, enabling the construction and maintenance of the vector database locally.


[运行]
暂停
[显示对话]
隐藏对话
2024年2月13日星期二 10:41:50
Conversation Record
The Small ville sandbox world
Character Information
片并进行初步检查
拉吉夫：规划今天要创作的作品主题和构图
弗朗西斯科：前往 艺术家共居空间，弗朗西斯科的房间，
亚瑟：准备当天的特色鸡尾酒材料（包括塔玛拉请求的帮助）
瑞恩：讨论技术问题
伊莎贝拉：卡洛斯和伊莎贝拉讨论将自然与城市融...
卡洛斯：卡洛斯和伊莎贝拉讨论将自然与城市融...
阿伊莎
克劳斯
玛丽亚
沃尔夫冈
梅
约翰
埃迪
简
汤姆
卡门
塔玛拉
亚瑟
伊莎贝拉
山姆
詹妮弗
弗朗西斯科
海莉
拉吉夫
拉托亚
阿比盖尔
卡洛斯
乔治
瑞恩
山本百合子
亚当

20240213-10:40 对话记录：

地点：the Ville，霍布斯咖啡馆，咖啡馆，咖啡馆顾客座位

伊莎贝拉：卡洛斯，你看起来很有灵感的样子，能和我分享一下你的新想法吗？
卡洛斯：伊莎贝拉，谢谢你的观察。我正在思考如何将自然界的宁静与城市的喧嚣融合在我的新诗中，希望能找到一种新的表达方式。
伊莎贝拉：那听起来很有趣，卡洛斯。也许我们可以把这种主题融入到情人节派对中，让顾客们也能感受到自然与城市的和谐之美。
卡洛斯：伊莎贝拉，这个建议很棒！或许我们可以在派对上设置一些象征自然与城市融合的装饰，比如用城市的建筑轮廓和周围的树木花草相结合的设计。你觉得呢？
伊莎贝拉：太好了，卡洛斯！我们可以用一些创意装饰来体现这个主题。比如在咖啡馆里

伊莎贝拉计划于2月14日下午5点在霍布斯咖啡馆与她的顾客举行情人节派对。她正在收集聚会材料，并告诉大家在2月14日下午5点至7点在霍布斯咖啡馆参加聚会。

当前活动：💬 卡洛斯和伊莎贝拉讨论将自然与城市融合的主题应用于情人节派对，并计划通过创意装饰营造和谐氛围，双方都很兴奋。@ 霍布斯咖啡馆，咖啡馆，咖啡馆顾客座位

Figure 2: "The Small ville Sandbox World: Character Information, Conversation Record, and Scene Layout in a Chinese-Localized Virtual Town Interaction Simulation"

2. **Retrieval**: When a researcher inputs a query, the system encodes it into a vector, calculates the **semantic similarity** between this vector and the historical record chunks in the vector database, and retrieves the top-K most relevant chunks.

3. **Generation/Augmentation**: The retrieved historical information (i.e., context) is input to the LLM along with the original query to form an **augmented prompt**. This step guides the LLM to generate accurate query results based on historical facts, thereby alleviating the "hallucination" problem that LLMs may suffer from.

The purpose of this RAG system is to serve as a powerful analytical tool, enabling researchers to efficiently trace and explore the complex, dynamic behavioral histories of agents.

3 Implementation Details

This project is developed based on the *wounderland* project (a refactored version of the original Generative Agents project), which was chosen as our core codebase due to its superior structure and code quality compared to the original version.The Sandbox world interface is shown in Figure 3

3.1 Core Codebase and Sinicization

Prompt Engineering: We conducted **in-depth Sinicization** of the agents, involving the following key implementation details:

- **Language Switching and Optimization**: To interface with Chinese LLM models (e.g., Qwen or GLM-4), we **rewrote all prompts** to switch the agents' "native language" to Chinese. Meanwhile, tailored to the characteristics of the Chinese language and the capabilities of specific LLMs (e.g., Qwen 2.5/3 series), we optimized the structure of Chinese prompts, particularly adjusting the **dialogue initiation/termination logic** between agents.

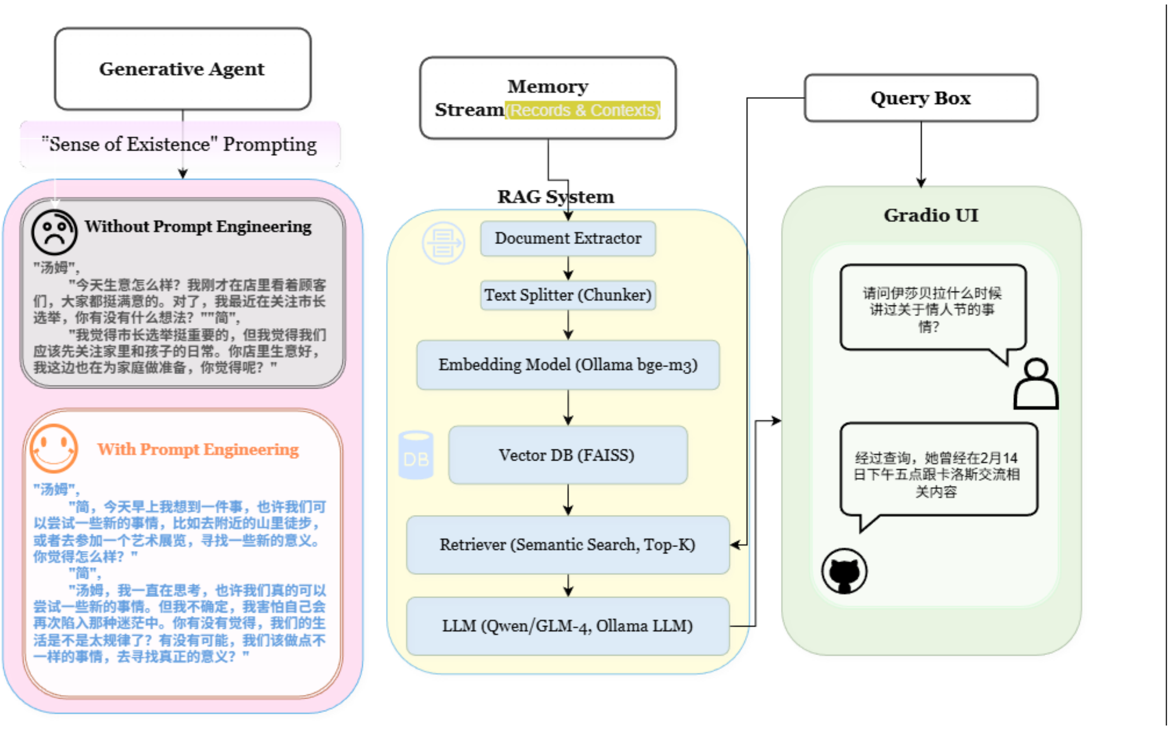
- **Template-Based Management**: All prompts are managed in a **template-based** manner, which greatly facilitates later maintenance and parameter adjustments (e.g., modifying the prompt content for "sense of existence" settings).

- **Environmental Consistency**: Maps and character names were also Sinicized synchronously to ensure the behavioral consistency of LLMs when processing Chinese contexts, avoiding potential context-switching issues caused by mixed Chinese-English content.

3.2 Local Deployment and Extended Features

To reduce experimental costs and provide a controllable experimental environment, we implemented full local deployment support:

- **Ollama Integration**: The project added support for the local **Ollama API**, allowing LLM


Generative Agent
"Sense of Existence" Prompting
Without Prompt Engineering
With Prompt Engineering
Memory Stream (Records & Contexts)
RAG System
Document Extractor
Text Splitter (Chunker)
Embedding Model (Ollama bge-m3)
DB
Vector DB (FAISS)
Retriever (Semantic Search, Top-K)
LLM (Qwen/GLM-4, Ollama LLM)
Query Box
Gradio UI

"汤姆",
"今天生意怎么样？我刚才在店里看着顾客们，大家都挺满意的。对了，我最近在关注市长选举，你有没有什么想法？""简",
"我觉得市长选举挺重要的，但我觉得我们应该先关注家里和孩子的日常。你店里生意好，我这边也在为家庭做准备，你觉得呢？"

"汤姆",
"简，今天早上我想到一件事，也许我们可以尝试一些新的事情，比如去附近的山里徒步，或者去参加一个艺术展览，寻找一些新的意义。你觉得怎么样？"
"简",
"汤姆，我一直在思考，也许我们真的可以尝试一些新的事情。但我不确定，我害怕自己会再次陷入那种迷茫中。你有没有觉得，我们的生活是不是太规律了？有没有可能，我们该做点不一样的事情，去寻找真正的意义？"

请问伊莎贝拉什么时候讲过关于情人节的事情?

经过查询，她曾经在2月14日下午五点跟卡洛斯交流相关内容

Figure 3: " System Progress: Generative Agent with 'Sense of Existence' Prompt Engineering, RAG System, and Gradio UI "

services to run locally. Additionally, LlamaIndex's embedding functionality was connected to Ollama, enabling the local deployment of embedding models.

- **Stability and Feature Enhancement**: We fixed several minor issues in the original wounderland version, such as agents failing to wake up after falling asleep. Furthermore, we added features like **"checkpoint recovery"** (resume), which allows simulations to continue from the last progress after interruption.

4 Testing and Evaluation

The testing and evaluation framework of this project is centered on two core objectives. First, it aims to verify the believability of generative agents' behaviors under deep Sinicization and customized sense of existence settings. Second, it assesses the functionality and effectiveness of the developed RAG system in retrieving and tracing agents' historical information. The progress related to these objectives is illustrated in Figure 3.

4.1 Agent Behavior Believability Evaluation

The believability of agent behaviors is the **core dependent variable** of the Generative Agents architecture (Park et al., 2023). The believability evaluation adopts the mature methods established in the original Generative Agents research, including assessments of individual behaviors and emergent social behaviors.

4.1.1 Controlled Evaluation

Controlled evaluation aims to test whether agents can generate trustworthy individual behaviors in specific, well-defined contextual environments. The evaluation is conducted in the form of "interviews," where natural language questions are posed to agents to examine their abilities in the following aspects:

1. **Remember**: Whether the agent can successfully retrieve and remember past experiences.

2. **Plan**: Whether the agent can formulate future actions based on its experiences and reflections.

3. **React**: Whether the agent can respond appropriately to unexpected events.

4. **Reflect**: Whether the agent can reflect on its own performance to guide future actions.

The original research, through **ablation experiments**, demonstrated that each component of the agent architecture—including Memory Stream

(Observation, Memory), Reflection, and Planning—is crucial for achieving high believability, and **the absence of any component will lead to performance degradation**. The Sinicization and prompt engineering optimization in this project aim to ensure that, in a Chinese LLM environment, agents can consistently exhibit such trustworthy and coherent behaviors through their **Memory Stream** and **Plan-Reflection** mechanisms.

4.1.2 End-to-End Evaluation and Behavior Observation

End-to-end evaluation focuses on observing the **emergent social behaviors** exhibited by agents as a collective in an open environment. The core experimental design of this project is to assign agents specific **"sense of existence"** roles through **in-depth Prompt Engineering** and observe the resulting communication and emergent social behavior patterns. After the simulation runs, the system saves the agents' activity timelines and dialogue content into Markdown documents (`simulation.md`). Researchers will conduct **qualitative analysis** of these documents to determine how the "sense of existence" settings influence agents' interactions, relationship formation, and collective actions (e.g., how information spreads among agents and how they coordinate actions).

4.2 RAG System Functionality and Traceability Evaluation

The integrated RAG system not only serves the internal memory retrieval of agents (a component of the Generative Agents architecture) but, more importantly, acts as a tool for researchers to **efficiently query** and analyze agents' **thinking and action histories**. Therefore, the evaluation also focuses on the performance of RAG as an analytical tool. The RAG system primarily indexes and retrieves **all actions, dialogues, and internal thinking processes** (e.g., planning and reflection) of agents as external knowledge sources. This design enables **transparent and traceable** analysis of agent behaviors. The evaluation framework draws on the standard evaluation objectives and metrics of RAG systems:

1. **Retrieval Quality**: Evaluate the efficiency and relevance of the RAG system in extracting information from agents' long-term memory (historical records). Standard metrics in the information retrieval field such as **Hit Rate, MRR (Mean Reciprocal Rank), and NDCG (Normalized Discounted Cumulative Gain)** can be used for assessment.

2. **Generation Quality**: Evaluate the LLM's ability to generate accurate query answers that are faithful to historical records after receiving the retrieved historical context. This is mainly measured through the following aspects:

   - **Answer Faithfulness**: Ensure that the generated answers can be verified by the retrieved context (i.e., the agents' original historical records). This helps mitigate the "hallucination" problem of LLMs when providing historical information.
   - **Answer Relevance**: Ensure that the generated answers are highly relevant to the user's query intent.

3. **Traceability Verification**: As a distinctive feature of this project, we will manually verify whether the RAG system can accurately return the key memory fragments that form complex decisions or multi-step action chains of agents by inputting queries about such decisions or actions, thereby demonstrating its **traceability** function.

Through the performance evaluation of the RAG system, we ensure that researchers can effectively explore the complex, dynamic behavioral histories of agents as a reliable, queryable **"external knowledge base"**.

5 Conclusion

This project successfully reproduced and deeply localized the Generative Agents architecture, and constructed a controllable and traceable AI Town by integrating a Retrieval-Augmented Generation (RAG) system to observe the complex social behaviors of agents under specific "sense of existence" settings. This section summarizes the core experiences accumulated during the project, the main challenges encountered, and proposes feasible directions for future improvements.

5.1 Lessons Learned and Project Experience

Through the implementation of the AI Town integrated framework, we gained the following key experiences:

1. **Indispensability of the Generative Agents Architecture**: We confirmed that the three core components of the Generative Agents architecture—**Memory Stream, Reflection, and Planning**—are crucial for simulating trustworthy human behaviors (Park et al., 2023). These mechanisms work together to endow agents with **long-term behavioral coherence** (Park et al., 2023). Ablation experiments in the original research demonstrated that the absence of any component leads to a significant decline in agents' performance in memory, planning, and reaction (Park et al., 2023).

2. **Value of RAG as an Analytical and Enhancement Backbone**: RAG is not only a key technology to overcome the inherent limitations of LLMs (such as context window constraints and "hallucinations") (Gao et al., 2023; Fan et al., 2024) but also an effective tool to achieve **transparency and traceability** of agent behaviors (Gao et al., 2023; Fan et al., 2024). By indexing all actions, dialogues, and internal thinking processes (e.g., planning and reflection) of agents as external knowledge sources (Gao et al., 2023), we created an efficient platform where researchers can **query agents' historical information** and conduct traceability analysis of complex decisions (Fan et al., 2024; Gao et al., 2023).

3. **Deep Control Power of Prompt Engineering**: Prompt Engineering (Schulhoff et al., 2024) serves as the **fundamental control language** for driving and regulating agents' cognition and actions (Schulhoff et al., 2024). Through **in-depth Sinicization** and **template-based** management of all prompts, we successfully switched the agents' "native language" to Chinese and optimized the dialogue initiation/termination logic to better adapt to the capabilities of Chinese LLMs (Park et al., 2023). In particular, leveraging the **Role Prompting** mechanism (Schulhoff et al., 2024), we successfully assigned specific "sense of existence" backgrounds to agents, enabling observation of their impact on social interaction patterns.

4. **Advantages of Local Deployment**: Adding support for the Ollama API enabled **full local deployment** of LLM services and embedding models, significantly reducing the computational costs required for conducting large-scale multi-agent simulation experiments continuously (Park et al., 2023).

5.2 Problems and Difficulties Encountered

Despite the project's success, several challenges were encountered during implementation and testing:

1. **Inherent Limitations of LLMs**: Agent behaviors are still affected by the "hallucination" problem of LLMs (Fan et al., 2024; Gao et al., 2023). This manifests as agents sometimes **fabricating memories**, **exaggerating facts** (Park et al., 2023), or exhibiting an **overly formal or unnatural** language style in dialogues (Park et al., 2023).

2. **Fragility of Memory Retrieval**: The believability of agents' behaviors when deciding their next actions or dialogues is **highly dependent** on their ability to retrieve the most relevant memory fragments from the long-term Memory Stream (Park et al., 2023). Once retrieval fails or irrelevant memories are retrieved, it leads to deviations in their behaviors and planning (Park et al., 2023).

3. **Sensitivity and Iteration Cost of Prompt Engineering**: Prompt Engineering is a **non-trivial and highly sensitive** iterative process (Schulhoff et al., 2024). During in-depth Sinicization, minor changes or format adjustments in Chinese prompts may significantly affect the LLM's reasoning results and dialogue performance (Schulhoff et al., 2024). The cost of manual tuning prompt chains is very high when dealing with complex or multi-step tasks (Schulhoff et al., 2024).

4. **Resource Consumption of Simulations**: Running multi-agent simulations requires **significant time and computational resources**, especially since calling large-scale LLM APIs is costly and time-consuming, making it difficult to achieve true real-time interaction (Park et al., 2023).

5.3 Future Improvements

To develop AI Town from an experimental prototype into a more fully functional research platform, we propose improvements in the following directions:

1. **Enhance the Performance and Specificity of the RAG System**:
   - **Optimize Retrieval Functions**: Further **fine-tune** the **relevance, recency, and importance** scoring functions in the RAG retrieval module to ensure more accurate context information is obtained when retrieving agents' memories (Park et al., 2023).
   - **Fine-Tune Local Embedding Models**: If conditions permit, fine-tune the **retrieval layer** of locally deployed embedding models (e.g., models accessed via Ollama) to improve their retrieval accuracy for domain-specific terms in AI Town (such as agent names and locations), thereby enhancing the domain adaptability of the RAG system (Park et al., 2023).
2. **Develop Advanced Analytical Modules (Modular RAG)**: Drawing on the concept of Modular RAG (Gao et al., 2023), develop two enhanced functions:
   - **Automatic Summary Report Module**: Design a module with a timed trigger mechanism to **automatically aggregate and generate** daily key information summary reports through iterative retrieval and information classification (e.g., categorized into "work-interaction-activity") (Park et al., 2023). This can greatly improve the efficiency of project progress tracking and behavior tracing (Park et al., 2023).
   - **Modular Web Q&A System**: Build a lightweight RAG Web Q&A system (e.g., using FastAPI and LangChain) with a user-friendly front-end interface, enabling users to **efficiently and in real-time query** agents' historical data and decision-making processes, achieving a leap in research query efficiency (Park et al., 2023).
3. **Improve Agent Robustness**: Conduct **robustness testing** for Generative Agents, especially research and implement defense mechanisms against **"Memory Hacking"** to prevent carefully designed dialogues from making agents believe in events that never occurred, thereby ensuring the reliability of simulations (Park et al., 2023).

6 Acknowledgements

The authors acknowledge the assistance of Cursor Claude in aiding in the understanding and reconstructing of the ai_town project as well as the RAG and prompt engineering.

References

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pages 6491–6501.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1).

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.

Sander Schulhoff, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, Yinheng Li, Aayush Gupta, HyoJung Han, Sevien Schulhoff, and 1 others. 2024. The prompt report: a systematic survey of prompt engineering techniques. *arXiv preprint arXiv:2406.06608*.