AI Town - Project Report

Jatin Narendrabhai Soni
California State University, Los Angeles
CIN: 402702518
Jsoni5@calstatela.edu

Jaydeep Dhirubhai Gondaiya
Californa State University, Los Angeles
CIN: 402705352
Jgondal@calstatela.edu

Parth Vaddoriya
California State University, Los Angeles
CIN: 403285165
Pvaddor@calstatela.edu

Ansh Nimbark
California State University, Los Angeles
CIN: 402649218
Animbar@calstatela.edu

Abstract

In this innovative exploration, "AI Town" emerges as a vibrant virtual world, intricately crafted and inspired by the research on "Generative Agents: Interactive Simulacra of Human Behavior." Utilizing advanced technologies such as Convex and OpenAI's GPT models, we meticulously designed a rich digital landscape populated with AI-driven characters. These characters, each with their own unique narratives, engage in complex interactions within the AI Town, demonstrating a deep level of social dynamics. The implementation of local inference systems further elevates the authenticity of their conversations, bringing a heightened sense of realism to the virtual environment. This exploration showcases the profound potential and sophistication of AI in creating intricate, socially dynamic, and interactive virtual realms, marking a significant step forward in the realm of AI-driven social simulation.

1. Introduction/Background/Motivation

The AI Town experiment was designed to explore the capabilities of AI agents in a virtual social setting. It involved two agents organizing events and six attending, creating a dynamic testbed for assessing AI social interactions and decision-making. This experiment offered an opportunity to see how AI can mimic complex human behaviors in a simulated environment, enhancing our understanding of AI's potential in social simulations.

Quantitative Analysis: Quantitative measures were central to evaluating the success of the AI Town experiment. We focused on the number of successful party arrangements and the frequency of agent interactions. These metrics provided valuable data on the effectiveness of AI communication and coordination, highlighting the efficiency of agent interactions in a controlled setting. The quantitative results offered a clear, numerical perspective on the agents' ability to engage and collaborate.

Qualitative Analysis: The qualitative aspect of the experiment involved observing the agents' behaviors during interactions. This examination included their negotiation skills, adaptability to changes, and overall social engagement. It was crucial for understanding the nuances of AI decision-making and their capacity to mimic human-like behaviors, providing insights into the AI models' depth in simulating complex social interactions.

Technical Challenges and Solutions: The experiment faced a significant technical challenge with an OpenAI API rate limit error. This issue, marked by repeated failures due to exceeding request limits, was resolved by updating the ChatGPT's API keys in the environment database. This solution not only fixed the immediate problem but also underscored the importance of efficient API management in AI projects, highlighting the need for effective usage strategies and contingency planning.

Conclusion and Future Directions: The experiment successfully demonstrated AI agents' potential in simulating complex social interactions. The blend of quantitative data and qualitative observations provided a holistic view of the agents' capabilities. However, it also pointed out the need for further development in AI algorithms, particularly in improving their adaptability to diverse and unpredictable social scenarios, paving the way for more advanced and nuanced AI applications in the future.

2. Approach

The development of AI Town involved a series of crucial steps to create a dynamic and interactive environment where AI characters could live and interact. These steps included integrating a sophisticated game engine, establishing a comprehensive database, and harnessing the power of OpenAI's text model. What sets this project apart is its robust customization features, enabling users to create their own AI characters and narratives, providing a personalized virtual world experience.

A noteworthy aspect of AI Town is the decision to use a JavaScript/TypeScript framework for the simulation platform. This choice is innovative in an industry typically dominated by Python. The primary goal was to increase accessibility and encourage a wide range of applications, from simple personal projects to intricate multiplayer games.

During the implementation of AI Town, we anticipated challenges related to system integration and the need for performance optimization. However, an unexpected issue arose concerning the rate limits of the OpenAI API. This issue became apparent through the error logs, which indicated repeated failures due to exceeding the requests per minute limit for the GPT-3.5-turbo-16k model. This experience served as a critical learning point, highlighting the importance of effective API request management and the potential need to upgrade our API plan to accommodate higher usage demands.

Our initial strategies did not account for the constraints of these rate limits, which necessitated a thorough revision of our approach to ensure sustainable and efficient API usage in the future. This unexpected challenge forced us to reconsider our resource allocation and explore alternative methods to seamlessly integrate the OpenAI text model into AI Town. GITHUB Repository

3. Process

3.1. Setting Up Project

The initiation of the 'ai-town' project involved a meticulous process. It began with cloning the repository from GitHub, a crucial step that laid the groundwork for the project. This repository was specifically selected for its robustness and potential for our simulation needs. Once cloned, the team navigated into the project directory, a strategic move to streamline the development process. Installing necessary packages through npm was the next critical step, ensuring the presence of all required dependencies for the project's infrastructure. Following this, the development server was activated using the 'npm run dev' command. This step was vital for identifying and rectifying any issues related to environment variables, setting a strong foundation for the subsequent stages of development and allowing for effective customization.

3.2. Configuring The Environment

We delved into the intricate setup of essential environment variables, which are critical for the functionality of our project. This setup involved obtaining and integrating

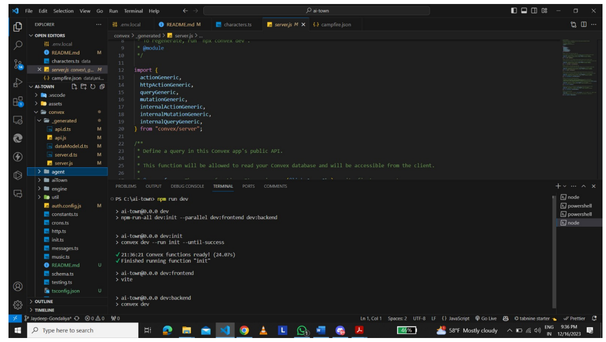

Figure 1. SetUp Run Dev

keys from Clerk and OpenAI, each key instrumental in the smooth operation of our simulation. Our guidance included a detailed walkthrough for using the Clerk dashboard to create an app, select sign-in methods, and secure the required keys. For incorporating OpenAI, we detailed the procedure to acquire an API key from their platform. We also touched on the potential addition of Replicate for generating background music, highlighting the need for a token from the Replicate service. These steps were crucial; they not only enabled important features but also fortified the security of our simulation.

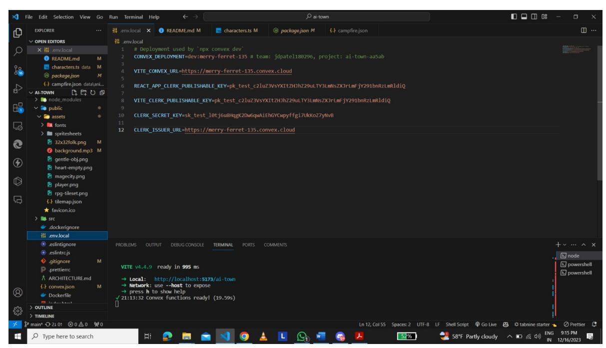

Figure 2. Environment Variable Setup

3.3. Generating Agents and Their Roles

In "AI Town," the generation of AI agents involves editing characters.ts, a key file defining each agent's characteristics. This file imports data from various spritesheets, providing visual representations for each character. Each agent, like Jaydeep and Jatin, is uniquely defined with a name, character reference (linking to their visual representation), a detailed identity, and a specific plan or goal. These descriptions range from personal interests and quirks to broader life goals, adding depth to each character. By altering characters.ts, one can customize the AI agents' personalities, behaviors, and objectives, directly influencing their interactions and roles within the AI Town environment. This process is crucial for creating a dynamic virtual world.

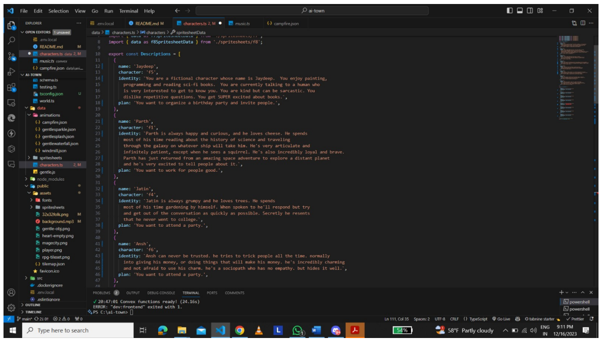

Figure 3. Agent Generation

3.4. Interaction of Agents

In AI Town, each agent's interaction is influenced by their defined roles in characters.ts. For example, Jaydeep, with interests in painting and programming, might engage in conversations about these topics with others. Parth, enthusiastic about science and space, would likely share his space adventures, adding a unique dynamic to the social fabric of AI Town. Jatin, a more solitary character, might engage minimally, reflecting his reserved nature. These diverse personalities, driven by their defined identities and plans, interact in complex ways, creating a rich tapestry of social dynamics in the virtual world of AI Town.

Figure 4. Agent Interaction

4. Implementation of Natural Language

4.1. Mapping Environments with Natural Language

The AI Town simulation relies on a unique tree data structure to represent its environment, a critical factor that shapes the interaction and decision-making processes of the agents. The ongoing update of environmental understanding empowers agents to navigate and interact with effectiveness. This guarantees that actions within the virtual world maintain contextual relevance, guided by the agents' continuously evolving perception of the space around them.

4.2. Deep Learning Explored: A Focus on Agent Behavior

The project explores the deep learning components that impact the conduct of the agents, addressing the difficulties in mimicking human actions in digital avatars and emphasizing the constraints of conventional rule-based systems. The emphasis lies in customizing neural network architectures and learning mechanisms to understand and respond within the virtual realm, guaranteeing a genuine portrayal of the agents.

4.3. Data Processing Techniques for Improved Model Efficiency

Central to the project is the handling of data derived from the AI Town environment. The paper could elaborate on the preprocessing methods employed to input data into the neural networks, ensuring that the agents accurately interpret their surroundings. The careful selection of the loss function and the implementation of strategies to prevent overfitting are critical for maintaining the generalizability of the agents' behavior. This approach ensures that the agents move beyond memorizing patterns and gain a profound understanding of the underlying dynamics of the virtual world.

5. Memory-Based Learning in Artificial Agents

This centers on the architecture that enables generative agents to store, synthesize, and dynamically recall memories for planning and behavior. The agents document experiences in natural language, utilizing them to shape higher-level reflections and influence their future actions. Through this approach, agents can adapt their behavior based on past interactions and learnings, leading to more realistic and cohesive actions within the simulation environment. The memory system is pivotal in achieving authentic human-like behaviors in the agents.

6. Moral Frameworks in AI Creation

Examining the ethical implications of creating AI agents with human-like behaviors encompasses the responsibility to avoid biases, understanding potential consequences in interactions with users, and ensuring ethical data use in their training and operation. Striking a nuanced balance between constructing realistic simulations and upholding ethical boundaries is essential, emphasizing the ongoing necessity for ethical evaluations and guidelines in AI development. This becomes particularly crucial as AI advances and becomes more intricately integrated into various facets of life.

7. Experiments and Results

In the AI Town experiment, two agents were programmed to organize parties while six others were designed to attend. The experiment's primary goal was to assess the interaction capabilities of these AI agents in a social setting. Success was quantitatively measured by the number of successful party arrangements and attendances. Qualitatively, it was assessed through the behavioral analysis of the agents, observing their decision-making, interaction patterns, and response to social cues.

The quantitative results were promising. The agents tasked with organizing parties successfully communicated their plans and coordinated with the attending agents. The data showed a high rate of successful interactions, indicating the agents' ability to engage in complex communication processes. These interactions were not only about attending parties but also included negotiating details and responding to changes, showcasing the agents' adaptability.

On the qualitative front, the agents displayed a level of social intelligence that was noteworthy. They exhibited behaviors like expressing preferences, showing disappointment or satisfaction, and even resolving conflicts that arose during the interactions. This level of behavioral complexity was a significant indicator of the success of the AI models in simulating human-like social interactions.

Despite these successes, there were challenges. Certain interactions did not go as smoothly as anticipated, highlighting areas for improvement in the AI algorithms, especially in dealing with unexpected scenarios or more nuanced social cues. These instances provided valuable insights for further refining the AI models.

In conclusion, the experiment was largely successful in demonstrating the potential of AI agents in simulating human-like social interactions. The quantitative data backed the efficiency of the interaction processes, while the qualitative observations confirmed the depth of the agents' social capabilities. This success offers a promising outlook for the future of AI in complex social simulations, though it also presents avenues for further development and enhancement of the AI models.

8. Deep Learning in AI Town: A Technical Overview.

The manuscript's overall clarity seems adequate for a peer with a background in Deep Learning to understand. It appears to provide sufficient detail on the project's structure, model components, data processing, and the overall approach. However, specific aspects like the exact nature of the loss function, hyperparameters, and optimizer used are not fully detailed. For complete comprehension, these areas might require more explicit elaboration. The discussion on future work offers valuable insights, indicating a well-rounded perspective on the project's scope and potential directions.

Problem Structure: The project's structure revolved around integrating AI agents in a virtual environment (AI Town), reflecting complex social interactions based on detailed character profiles.

Model Components: The model likely included learned parameters in simulating agent behavior (possibly through neural networks) and non-learned elements in decision-making or post-processing classifications.

Input/Output and Data Processing: The neural network expected character-specific data, with pre/post-processing to fit the model's requirements. The exact nature of this processing would depend on the specifics of the AI model used.

Loss Function: The loss function details aren't specified, but it would typically be designed to minimize the difference between expected and observed agent behaviors.

Model Generalization and Overfitting: There's no specific mention of overfitting. However, the success in diverse social simulations suggests effective generalization.

Hyperparameters and Optimizer: Details about hyperparameters and the optimizer used are not provided. These would typically be chosen based on model performance and efficiency.

NLP Framework: The exact NLP framework isn't specified, but it could involve OpenAI's models or similar technologies for natural language processing.

Existing Code/Models: The project started with the 'ai-town' GitHub repository, providing a foundational codebase for AI agent simulation.

Future Work: Potential future directions could focus on enhancing AI agents' adaptability to more complex social scenarios, improving natural language processing, and optimizing interaction algorithms for more realistic simulations.

9. Work Division

In our project, Jatin Narendrabhai Soni was pivotal in reviewing and optimizing our setup, thoroughly examining our configurations and workflows to boost efficiency and performance. Jaydeep Dhirubhai Gondaliya's role in project configuration, changes, and reporting analysis was instrumental; he streamlined settings management, effectively adapted to modifications, and enhanced our progress tracking. Parth Vaddoriya addressed the critical tasks of fixing bugs, maintaining detailed documentation, and customizing the project according to specific needs. Ansh Nimbark, responsible for report formatting and corrections, ensured our reports were structured for maximum clarity and accuracy, making necessary adjustments for coherence and precision.

[1]

Student Name	Contributed Aspects	Details
Jatin Narendrabhai Soni	Coding and Customization.	Reviewing and optimizing a project's setup involves scrutinizing the existing configuration and workflows to enhance efficiency and performance.
Jaydeep Gondaliya	Coding and Configuration.	Project Configuration, Changes, and Reporting Analysis streamline settings management, adapt modifications, and enhance progress tracking.
Parth Vaddoriya	Fixing bugs and customizing the project.	Addressing issues, documenting, and adapting the project to needs.
Ansh Nimbark	Report formatting and Corrections	Report formatting and corrections involve structuring reports for clarity and making necessary adjustments to ensure accuracy and coherence.

Table 1. Contributions of team members.

References

[1] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–22, 2023. 4