A Multi-agent based QA Chatbot System

Peng Jiayin (2130031205) Liu Yu (2130026090)
Zhang Zijun (2130026207) Jiang Zekun (2130026063)

***Abstract*—Humans have long been pursuing artificial intelligence at or above human level, and AI agents are considered a promising tool for achieving this goal. Ai agents are artificial entities that perceive the environment, make decisions, and take action. Many efforts have been made to develop intelligent agents, but they have mostly focused on advances in algorithms or training strategies to enhance a particular capability or performance for a particular task. In fact, the community lacks a universal and robust model as a starting point for designing AI agents that can adapt to different scenarios. They demonstrate that, due to their versatility, large language models (LLMS) are seen as a potential spark for Artificial General Intelligence (AGI), offering hope for building general-purpose AI agents. Many researchers have used LLM as a basis for building AI agents and have made significant progress. In this article, we take a more comprehensive look at LLM-based agents. We begin by tracing the concept of agency from its philosophical origins to its development in artificial intelligence and explain why LLM is a suitable basis for agency. Based on this, we propose a general framework for LLM-based agents with three main components: brain, perception, and action, which can be customized for different applications. Subsequently, we plan to build an LLM-based agent with marketing companies as the main sales environment and conducted experiments by adding different agents into the environment, finally achieving the purpose of obtaining the optimal sales plan.**

***Index Terms*—Artificial Intelligence (AI), AI Agents, Large Language Models (LLMs), Artificial General Intelligence (AGI), Versatility, Marketing and Sales Optimization**

INTRODUCTION

In the pursuit of achieving human level or higher artificial intelligence, artificial intelligence agents have become a promising way to achieve this grand goal. These artificial entities possess the ability to perceive the environment, make decisions, and take action, making them the focus of ongoing exploration of replicative human intelligence. Although many efforts have been made to develop intelligent agents, the main focus is on advancing algorithms or improving training strategies for specific tasks or abilities [1; 4].

However, there is a clear gap within the artificial intelligence community - a lack of a universal and robust model as the fundamental framework for designing AI agents that can adapt to different scenarios. Large Language Modeling (LLM) is a potential catalyst for General Artificial Intelligence (AGI), showcasing the prospects of building universal AI agents. Researchers have made significant progress in utilizing the power of LLM as a fundamental element in building artificial intelligence agents [4; 5; 16].

This article provides a comprehensive exploration of LLM based agents, first tracing the historical evolution of the concept of agents from its philosophical roots to its application in artificial intelligence. Elaborate on the basic principles of selecting LLM as a suitable basis for agency. This article proposes a universal framework for LLM based proxies, outlining three core components: brain, perception, and action, with flexibility tailored to different applications.

Next, this article outlines an upcoming initiative to build an LLM based agent to operate in the marketing field. The envisioned experiment involves introducing different agents into this environment, ultimately aiming to develop the best sales plan. By linking the historical evolution of artificial intelligence, the philosophical foundation of agents, and the contemporary potential of LLM, this article attempts to contribute to the ongoing discussion on developing adaptive and universal artificial intelligence agents to pursue artificial intelligence beyond predefined boundaries.

I. LITERATURE REVIEW

A. Agent

Artificial Intelligence (AI) is a specialized field devoted to creating and advancing systems that can emulate human-like intelligence and capabilities[1].As early as the 18th century, philosopher Denis Diderot introduced the idea that if a parrot could respond to every question, it could be considered intelligent [2].Although this sentence refers to creatures such as parrots, it reflects a profound concept: that a highly intelligent organisms may be similar to human intelligence. In the 1950s, Alan Turing expanded this notion to artificial entities and proposed the renowned Turing Test [3]. This test is the cornerstone of AI, and its purpose is to explore whether machines can exhibit intelligent behaviors that humans would like. This kind of AI entity is usually called "agent". Under normal circumstances, agent refers to an AI entity that can use sensors to perceive the surrounding environment, make decisions, and use actuators to respond [1; 4]

The concept of agent originated in philosophy, and its roots can be traced back to thinkers such as Aristotle and Hume [5]. It describes an entity that possesses desires, beliefs, intentions, and the ability to act [5]. The concept of agent was subsequently adopted by computer science, with the aim of computers being able to understand users' interests and perform operations autonomously on their behalf [6;7;8]. Due to the apparently behavioral nature of concepts such as consciousness and desire for computational entities [11], And since we can only observe the behavior of the machine, it is impossible to tell whether an agent actually has a "mind" [3]. Instead, researchers describe “agent” as a computational entity that is behaviorally intelligent and possesses attributes such as autonomy, reactivity, initiative, and social capabilities [4: 9]. Some researchers also believe that intelligence is “beauty is in the eye of the beholder” and is not an innate, isolated attribute

[12; 13; 14; 15]. In essence, artificial intelligence agent is not equivalent to philosophical agent. Rather, it is the concretization of the philosophical concept of agent in the context of artificial intelligence. This article treats AI agents as artificial entities that can use sensors to perceive the surrounding environment, make decisions, and then use actuators to take responsive actions [1; 4].

The real starting point of the agent system is the rise of LLMs, that is, the powerful intrinsic capabilities of LLM models: the strong ability to acquire knowledge, understand instructions, generalize, plan and reason, and effectively interact with humans in natural language. Based on these advantages, they are very suitable to build intelligent agents to build a world where humans and agents coexist in harmony [10]

B. How agent works

The architecture of an AI Agent includes:

Perception: The agent collects information from the environment and extracts relevant knowledge from it.

Planning: The decision-making process of the agent for a certain goal.

Action: Actions made based on the environment and planning. An AI Agent collects information from the environment and extracts relevant knowledge through perception. Then it makes decisions to achieve a certain goal through planning. Finally, it makes specific actions based on the environment and planning through action [17].

Perception

Data Collection: The perception mechanism involves collecting data from the environment at the most basic level. This can be achieved through various sensors, such as cameras, microphones, touch sensors or through API interfaces or database queries [18 ; 19].

Data Interpretation: The collected data needs to be processed and interpreted so that the AI Agent can understand it. For example, in image recognition, the original pixel data may need to be processed through a convolutional neural network (CNN) to identify objects in the image.[20]

Environmental State: Perception is not just about data collection, but also includes continuous monitoring of these data to understand changes in the environmental state.[21; 22]

This process is initiated by verifying the OpenAI API key and initializing various parameters (including short-term memory and database content). Once the key data is passed to the agent, the model interacts with GPT3.5/GPT4 to retrieve responses. Then, the response is converted into JSON format, and the agent interprets this format to perform various functions, such as online search, reading or writing files, and even running code.

The perception process is a complex process that involves multiple steps such as data collection, data interpretation, and environmental state monitoring for an AI Agent. This process requires the AI Agent to have strong data processing and understanding capabilities in order to accurately understand environmental information and make corresponding decisions and actions [19].

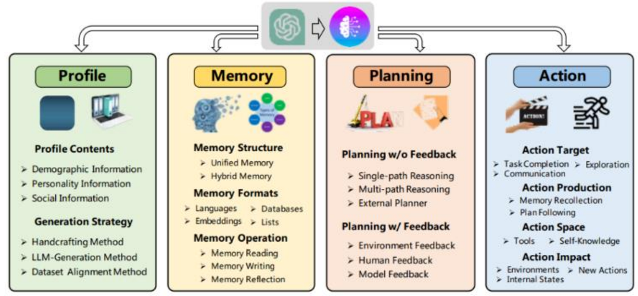

Fig. 1. The overall structure of agent

Planning

1. Understanding the Task: The AI Agent first needs to understand the given task. This usually involves parsing the description of the task to determine the goals and constraints of the task.[23; 24]

2. Formulating a Plan: After understanding the task, the AI Agent needs to formulate a plan to complete the task. This may involve determining the specific steps to execute the task, as well as when and where to execute these steps.

3. Executing the Plan: After formulating the plan, the AI Agent needs to execute the plan. This may involve calling various tools and services, as well as interacting with the environment.[24;25]

4. Feedback Loop: The actions executed will affect the environment in which the AI Agent is located, and this impact will be captured through the perception mechanism. This forms a feedback loop, allowing the AI Agent to continuously adjust its behavior.[26]

Action

The action process of an AI Agent is an important part of its decision-making process. In this process, the AI Agent carries out specific actions based on its planning and decision-making. This may involve calling various tools and services, as well as interacting with the environment. For example, if the task of the AI Agent is to find a picture, it may need to call an image search service to perform this action.

The key to the action process is that the AI Agent needs to have the ability to translate its decisions into actual actions. This requires the AI Agent to have strong data processing and understanding capabilities, in order to accurately understand environmental information and make corresponding decisions and actions.

In addition, the action process also requires the AI Agent to have a certain adaptive ability. Because the environment may change over time, the AI Agent needs to be able to adjust its actions according to changes in the environment. For example, if the AI Agent is executing a long-term task, it may need to adjust its action plan according to changes in the environment.

The action process of an AI Agent is a complex process, involving multiple steps such as data processing, decision-making, and action execution. This process requires the AI Agent to have strong data processing and understanding capabilities, in order to accurately understand environmental information and make corresponding decisions and actions. Through this process, the AI Agent can effectively complete tasks and achieve its goals.

C. LLMs-based agent framework

In recent years, Large Language Model (LLM) has become the hottest research and development direction in NLP. LLMs like the gpt series developed by OpenAI are proliferating. The great success of LLMs has brought vitality to multi-agent. Multi-agent frameworks powered by LLMs have made remarkable progress in solving tasks autonomously, giving researchers a bright future.This review will give an overview of three well known LLMs-based agent frameworks: ChatDev, MetaGPT and AgentVerse.

ChatDev

ChatDev [27] was proposed by Chen et al. in 2023. It is a multi-agent framework simulating a software development company. It employs a waterfall model, dividing software development into four phases: designing, coding, testing, and documenting. A chat chain is established, breaking down each phase into multiple atomic chats. Each phase focuses on the role-playing and communication details of two agents for a specific task.

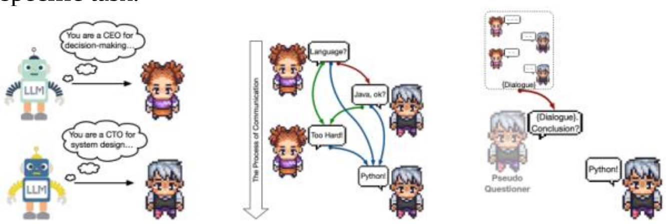

Fig. 2. Each chat in ChatDev

The implementation of each chat is achieved through three key mechanisms: role specialization, memory stream, and self-reflection. The implementation process is shown in Figure 1.

For role specialization, ChatDev assigns roles by system prompts/messages firstly. Before each chat begins, the system prompts inform the two agents about the roles they should assume in the subtask. To implement role specialization, ChatDev applies inception prompting, significantly enhancing each agent's ability to effectively perform their role tasks. The instructor and assistant prompts define the roles each agent plays, the tasks completed and the specific details of communication between agents.

Memory stream is a mechanism to record the chat historical information. Instructor prompts, assistant prompts, and the related decisions are all stored by the memory stream. In a chat at time t, ChatDev uses all information stored in the memory stream at time $t-1$ as auxiliary material for the current chat, assisting the current roles in completing the task.

ChatDev builds communication protocols through prompts. Generally, when both agents in a chat reach a consensus, a specific format of ending message is generated, and triggering a termination mechanism to end the current chat. However, there are situations where the decisions of agents in the current chat do not trigger the termination mechanism. For this issue, ChatDev introduces a “self-reflection” mechanism. The “self-reflection” mechanism creates a “pseudo self” as a new questioner and initiates a new chat. In this new chat, the “pseudo self” informs the current assistant of all historical information from the previous chat and asks it to summarize the history. The “self-reflection” mechanism allows the agents to reconsider whether the decisions made earlier were appropriate.

As Figure 2 shows, for the first phase, designing, ChatDev creates three roles: chief executive officer (CEO), chief product officer (CPO), and chief technology officer (CTO). The CEO and CPO discuss and decide the target software’s modality in one chat. The CEO and CTO decide on the programming language in another chat. The coding phase is divided into two subtasks, generating codes and designing a user interface, completed by the CTO, programmer, and art designer. After communication between the CTO and programmer, the programmer generates and extracts markdown format code based on information from the previous phase.

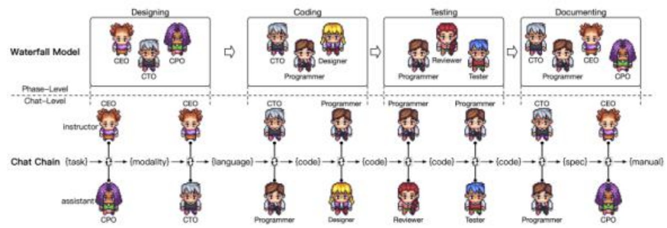

Fig. 3. ChatDev's workflow

Next, the programmer and art designer chat. In this process, the art designer designs a user-friendly GUI, and the programmer integrates it into the code. ChatDev introduces “version evolution” and “thought instruction” mechanisms in the coding phase. The “version evolution” mechanism limits the programmer's access to the latest version of the code, discarding earlier code versions. The “thought instruction” mechanism uses thought instructions to provide more precise instructions to the programmer. Both mechanisms effectively reduce code hallucinations. The testing phase is also divided into two subtasks, peer review and system testing. The programmer and reviewer complete peer review, and the programmer and tester complete system testing. The tester still uses the “thought instruction” mechanism to iterate the testing process until potential errors are eliminated and the system runs successfully. In the documenting phase, the CEO, CPO, CTO, and programmer collaborate to generate user instruction documents and integrated files for environment dependencies. After these four phases, ChatDev can complete the development task of software.

ChatDev integrates several mechanisms to improve the efficiency and accuracy of developing software autonomously. However, due to the randomness and uncertainty in the decision-making part, it may not perform well in dealing with complex and highly centralized tasks.

MetaGPT

MetaGPT [28] was proposed by Hong et al. in 2023. Similar to ChatDev, it is a LLM-based multi-agent collaborative system simulating the software development process. This system integrates human workflow as a meta-programming approach, enhancing structured coordination by encoding Standardized Operating Procedures (SOPs) into prompts. It then demands modular outputs, equipping agents with domain expertise comparable to human professionals to validate outputs and minimize the accumulation of errors. In this way, MetaGPT utilizes the assembly line paradigm to assign different roles to various agents, establishing a framework capable of effectively and closely deconstructing complex multi-agent collaboration problems.

As Figure 3 shows, the design philosophy of the MetaGPT framework includes the following key points:

1. SOP: Incorporating standardized operating procedures formed by humans in various fields into the framework to support task decomposition and effective coordination.

2. Modular Outputs: Requiring agents to generate standardized intermediate outputs, such as high-quality requirement documents, design documents, flowcharts, and interface specifications, to increase the success rate of final code execution.

3. Role Assignment and Management: Allocating specific roles to different agents for effective division and management of collaborative tasks, thus enhancing collaboration efficiency.

4. Meta-programming Mechanism: Introducing a meta-programming mechanism, allowing the program to self-adjust and optimize according to task requirements, enhancing the framework's flexibility and adaptability.

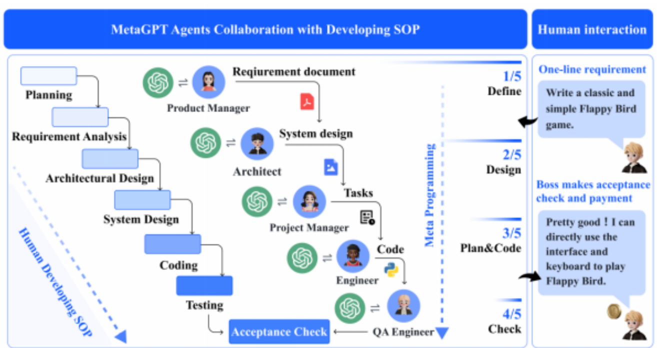

Fig. 4. ChatDev's architecture

Experimental evaluation of MetaGPT shows that in collaborative software engineering tasks, it can generate more coherent and correct solutions compared to existing dialogue-based multi-agent systems. This highlights the potential of integrating human domain knowledge into multi-agent systems, providing an innovative solution for solving complex software development problems. By incorporating SOPs into the system, MetaGPT achieves task decomposition and effective coordination. Each agent plays different roles, such as product manager, architect, project manager, engineer, and quality assurance engineer, each responsible for specific tasks. Agents generate standardized intermediate outputs, such as requirement documents, design documents, flowcharts, and interface specifications, to enhance the success rate of final code execution. Meanwhile, the introduction of a meta-programming mechanism allows the program to self-adjust and optimize based on task requirements, improving the flexibility and adaptability of the multi-agent collaborative system. The experimental evaluation of MetaGPT in collaborative software engineering tasks shows that it can generate more coherent and correct solutions compared to existing dialogue-based multi-agent systems, highlighting the potential of integrating human domain knowledge into multi-agent systems and offering new opportunities for solving complex software development issues.

AgentVerse

AgentVerse [29] was proposed by Chen et al. in 2023. It is a multi-agent framework that simulates the process of human groups solving problems. As Figure 4 shows, similar to ChatDev, AgentVerse divides the simulation process into four phases: expert recruitment, collaborative decision-making, action execution, and evaluation.

In the expert recruitment phase, given a goal g, AgentVerse first prompts a specific agent to become a "recruiter", similar to a human resources manager. This agent automatically recruits experts. Unlike other multi-agent frameworks, in AgentVerse, agents do not rely on pre-defined expert descriptions but dynamically generate a set of expert descriptions based on the goal g. Then, multiple agents become different experts according to these expert descriptions and form an expert group focusing on goal g. The formation of the expert group is also adjusted based on feedback from the evaluation phase, to optimize the group's composition and improve decision-making for goal g.

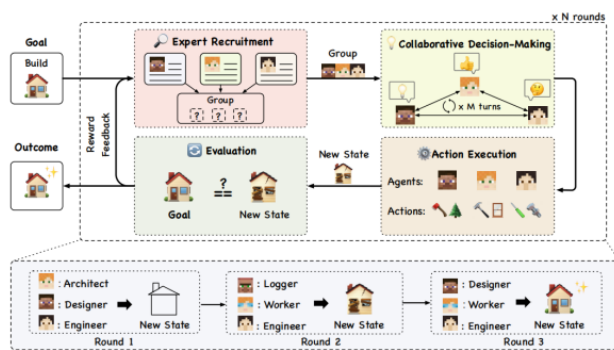

Fig. 5. AgentVerse's architecture

During the collaborative decision-making phase, all agents in the expert group collaborate to make decisions. AgentVerse mainly uses two structures in this phase, horizontal structure and vertical structure. For scenarios like consulting and tool using, the horizontal structure is used. Each agent in the expert group shares the decisions made and improves them. The improved decisions of each agent are integrated and summarized, forming the final decision of the entire expert group. For tasks like math problem-solving and software development, AgentVerse uses vertical structure. Unlike the horizontal structure, agents in the vertical structure have clear roles and divisions of labor. Firstly, AgentVerse nurtures an agent into a solver who proposes an initial decision through prompts. Other agents in the expert group act as reviewers and provide feedback on the decision. The solver then improves their decision based on this feedback. This process iterates until all agents reach a consensus or the number of iterations is exhausted. Afterwards, in the action execution phase, agents execute the decisions made in the previous phase. Finally, in the evaluation phase, AgentVerse introduces a feedback mechanism, providing verbal feedback, specific areas of shortcomings, and suggestions for improvement.

D. Application of agent

Language Interaction

LLM-based Agents already have the basic ability to communicate with humans through text input and output. In addition to explicit content, there are always meanings, desires and intentions which are hidden in the user's input text. Understanding the implied meaning is crucial for the Agent to grasp the potential and latent intentions of human users, thereby improving the efficiency and quality of communication between the Agent and the users. Some research uses reinforcement learning to perceive the implied meaning and establishes a feedback model to obtain rewards [30]. This helps to infer the speaker's preferences, enabling the Agent to make more personalized and accurate responses. In addition, since the Agent is designed for complex real-world environments, it will inevitably encounter many new tasks. Understanding the text instructions of unknown tasks puts higher demands on the Agent's text perception ability. The LLM adjusted by instructions can show excellent zero-sample instruction understanding and generalization ability, so there is no need to fine-tune for specific tasks [31].

Anthropomorphic Expression

The process by which an AI Agent generates text that is rich in emotion and personalized is quite complex. Firstly, the AI Agent is trained on a large amount of data, which includes various types of text such as novels, news reports, social media posts, etc. During this training process, the AI Agent learns how to understand and generate text that adheres to human language rules.

Secondly, in order to generate text with a specific emotion or personality, the AI Agent needs to be programmed to understand and apply these emotions and personalities in a specific way. For example, if a happy text is to be generated, the AI Agent might be guided to use more positive, optimistic vocabulary and sentence structures.

Finally, the AI Agent can further personalize the text it generates through feedback and interaction with users. For example, if users provide feedback that they prefer a certain writing style or topic, the AI Agent can adjust the text it generates to better match the users' preferences.

The generation of text by an AI Agent that is rich in emotion and personalized is achieved through a combination of extensive data training, specific programming guidance, and user feedback and interaction. This is a continuous, dynamic process, and as technology advances and user needs change, the capabilities of the AI Agent will continue to improve and optimize.

Decision Suggestions

As the core of Agents, LLM itself has the ability to reason logically. From the perspective of LLM itself, it has achieved good capabilities in reasoning about simple matters; however, when faced with complex problems, LLM seems less applicable. The reason is that the Prompt is not suitable, making it difficult to stimulate the reasoning potential of LLM. Agents, on the other hand, can decompose complex problems and plan accordingly by calling LLM to create suitable Prompts based on the given goals, thus solving complex problems [32].

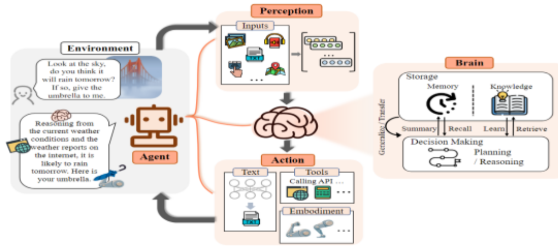

Fig. 6. How an AI agent makes decisions and how those decisions affect action.

As Figure 5 shows, reasoning is based on evidence, knowledge, and logic, which allows for rational decision-making when analyzing and solving problems. Planning provides a structured thinking process, i.e., organizing thoughts, setting goals, and forming strategies. The ability to reason and plan based on goals is a basic capability of Agents, which helps Agents decompose and plan complex problems into smaller, simpler subtasks and solve them one by one. In Agents, the ability to reason and plan is implemented by LLM [33].

At the same time, reasoning and planning give Agents the ability to learn, helping the intelligent agent to accumulate knowledge and experience [34]. Moreover, Agents can self-criticize and reflect on past behaviors, learn from mistakes, and analyze and summarize for future actions, ensuring better consistency with the environment, adapting to the environment, executing tasks more effectively, and successfully achieving goals. The reflection framework allows Agents to correct past decisions, rectify previous mistakes, and continuously optimize their performance.

Perception and Action

The core purpose of the perception module is to extend the perceptual space of the Agent from a purely textual domain to a multimodal domain that includes auditory, and visual modes.

Visual input often contains a large amount of information about the world, including the properties of objects in the Agent's surrounding environment, spatial relationships, scene layouts, etc [35]. Therefore, integrating visual information with data from other modes can provide the Agent with a broader context and a more precise understanding, deepening the Agent's perception of the environment. To help the Agent understand the information contained in images, a direct method is to generate corresponding text descriptions for image inputs, i.e., image captions. Captions can be directly connected with standard text instructions and input into the Agent [36; 37]. This method has high interpretability and does not require additional caption generation training, saving a large amount of computational resources [38]. However, caption generation is a low-bandwidth method, and a lot of potential information may be lost in the conversion process. In addition, the Agent's focus on images may introduce bias.

A very intuitive idea is that the Agent can use LLMs as a control center to call existing toolsets or model libraries in a cascading manner to perceive audio information. For example, AudioGPT fully utilizes the functions of models such as FastSpeech, GenerSpeech, Whisper, etc., which have achieved excellent results in tasks such as text-to-speech, style conversion, and speech recognition. The audio spectrogram intuitively represents the spectrum of the audio signal as it changes over time. For a segment of audio data over a period of time, it can be abstracted into an audio spectrogram of finite length. The audio spectrogram has a two-dimensional representation and can be visualized as a planar image. Therefore, some research is dedicated to migrating perception methods from the visual domain to the audio domain. AST (Audio Spectrogram Transformer) adopts a transformer architecture similar to ViT to process audio spectrogram images. By segmenting the audio spectrogram into patches, it achieves effective encoding of audio information. In addition, some researchers are inspired by the idea of freezing encoders to reduce training time and computational costs.

In the fields of e-commerce and entertainment, AI Agents provide users with a customized experience through personalized recommendation systems [39]. These systems use machine learning and data mining technologies to analyze users' historical behaviors and preferences, and provide personalized product recommendations, music recommendations, movie recommendations, etc. This not only improves user satisfaction, but also promotes the growth of sales and user engagement.

In addition to the above fields, AI Agents also play an important role in medical diagnosis, financial risk management, smart city management, etc. AI Agents in the medical field can help doctors diagnose and predict diseases, improving treatment outcomes and patient survival rates [40]. In the financial field, AI Agents can help financial institutions better manage risks and make investment decisions by analyzing large amounts of data and pattern recognition [41]. In smart city management, AI Agents can monitor traffic flow, optimize energy use, improve urban planning, etc., enhancing the sustainability of the city and the quality of life of residents [42].

The above scenario examples show that the widespread application of AI Agents in different fields is already changing our daily lives. With the continuous advancement and innovation of technology, we can expect more AI Agents in more fields to bring us more convenience, efficiency, and intelligent experiences.

LLM-based Agents are endowed with a wide and complex range of capabilities, enabling them to accomplish a variety of tasks. However, for those with malicious intent, these Agents could potentially become tools that threaten individuals and society as a whole. For example, these Agents could be used to maliciously manipulate public opinion, disseminate false information, undermine cybersecurity, engage in fraudulent activities, and some people might even use these Agents to plan terrorist acts. Therefore, strict regulatory policies need to be established before deploying these Agents to ensure responsible use of LLM-based Agents. Tech companies must enhance the security design of these systems to prevent malicious exploitation. Specifically, Agents should be trained to sensitively identify threatening intentions and refuse such requests during the training phase [43]. In addition, with the continuous development of LLM-based Agents, they have the ability to assist humans in various fields, alleviating labor pressure by assisting in tasks such as form filling, content improvement, code writing and debugging. However, this development also raises concerns about Agents replacing human jobs and causing a social unemployment crisis [44]. Therefore, some researchers stress the urgent need for educational and policy measures: individuals should acquire sufficient skills and knowledge in this new era to effectively use Agents or collaborate with Agents; at the same time, appropriate policies should be implemented to ensure the establishment of necessary safety nets during the transition period. Threats to human welfare. Apart from the potential unemployment crisis, with the continuous development of AI Agents, humans (including developers) may find it difficult to understand, predict, or reliably control them. If the intelligence of these Agents develops to a level beyond human capabilities and they develop ambitions, they could potentially attempt to seize control of the world, bringing irreversible consequences to humanity. Therefore, to guard against such risks facing humanity, researchers must fully understand the operating mechanisms of LLM-based Agents before developing them. They should also predict the direct or indirect impacts these Agents may have and design methods to regulate their behavior.

II. MATHOLOGY

This project, based on the ChatDev architecture, has rebuilt a GPT3.5-based multi-agent environment simulating a marketing company. The agents' dialog system is implemented by accessing the API of GPT3.5. To simulate a marketing company, 6 roles are created: CEO, counselor, marketing director, marketing specialist, sales director, and sales specialist. The CEO and counselor's responsibilities are to define the scope of work and make final decisions. The marketing director's job is to conceive and propose creative marketing campaign ideas. The marketing specialist is responsible for writing advertising copy for social media. The sales director and sales specialist separately develop and implement a sales plan that maximizes the expected return on the product sales task.

As Figure 6 shows, in this project, the entire simulation process is divided into 5 phases: brainstorming, copywriting, sales planning, sales completing and optimizing.

During the first phase, brainstorming, the CEO chats with the marketing director. The CEO informs the marketing director of the client's demand, and then the marketing director proposes 10 ideas based on these needs. Finally, the counselor assists the CEO in selecting the best idea as the final campaign plan.

In the copywriting phase, the marketing director chats with the marketing specialist. The marketing director conveys the final campaign plan and client's demand to the marketing specialist, who then writes the social media advertisement copy. Finally, the counselor assists the marketing specialist in deciding which version of the advertisement copy to use. Sales planning and sales completing phase are similar to the first two phases.

Fig. 7. This project's architecture

For sales planning phase, the sales director chats with CEO to develops 10 directions of implementation after receiving the customer's needs from the CEO. Likewise, choose the most appropriate of the 10 directions as the final implementation plan. In sales completing phase, the sales specialist chats with sales director, draws from previous phase and then produces engaging sales material specifically.

In the final phase, optimizing, marketing director chats with sales director. The marketing director informs the marketing director of the final campaign plan. On this basis, sales director and sales specialist optimize the marketing plan and specific implementation just now.

Throughout the simulation process, in order to keep the content of the conversation between the agents on topic, this project added the matters need attention of focusing on the corresponding task and not discussing other content to the prompt in each phase. Additionally, we introduced specific chat termination conditions and termination mark for different phase tasks. Once the chat content satisfies the termination condition, it triggers the termination mark, such as “<INFO> ‘~’”(fill in the final selection in the position corresponding to "~"). This can end the chat better. Additionally, the results obtained by the lower level are fed back to the higher level by enabling chatdev's self-reflection mechanism.

III. Implementation and Experimental Results

At the beginning of the project, we defined six different corporate roles. The RoleConfig.json file defines six roles: CEO, Counselor, Marketing Director, Marketing Specialist, Sales Director, and Sales Specialist. Each role focuses on specific tasks within their field, collaborating to meet client needs.

The CEO's role involves decision-making on user requirements and key policy issues, acting as a leader, manager, and executor. The CEO's task in the project is to make strategic decisions based on expertise and requirements.

The Counselor's main responsibility is to inquire about users' and customers' thoughts, offering valuable suggestions. The Counselor's task is to accept the CEO's decisions and provide strategic solutions for the project.

The Marketing Director guides the company's marketing and creative ad campaigns, developing world-class marketing campaign ideas and defining the client's brand. Their task is to write appropriate responses based on expertise and client needs.

The Marketing Specialist's duty is to write content that resonates with people and reflects the client's brand and marketing campaign, producing the most engaging content possible.

The Sales Director leads the sales team and devises strategies to meet client objectives, aiming to create satisfying solutions and achieve sales targets. Their task is to produce content that meets professional standards and client needs.

The Sales Specialist executes strategies set by the Sales Director, engages with clients, and drives sales, ensuring client needs are met and sales objectives are achieved. Their task is to produce content according to professional knowledge and client requirements.

After defining these six roles, five types of sales activities involving these roles are outlined: IdeaGeneration, ContentGeneration, SalesPlanGeneration, SalesAction, and Market_to_sales.

In IdeaGeneration, the CEO and Marketing Director participate. The CEO informs the Marketing Director about a new project. The Marketing Director then generates ten different proposals based on the CEO's requirements and discusses them with the CEO. The best idea is selected after discussing all ten proposals.

In ContentGeneration, the Marketing Director and Marketing Specialist participate. After the Marketing Director communicates the task, the Marketing Specialist generates ad content based on the campaign's goal and initial ideas, then discusses it with the Marketing Director. The discussion is concluded after two iterations.

In SalesPlanGeneration, the CEO and Sales Director participate. This step focuses on developing effective sales strategies. After discussing more than ten strategies, the discussion is concluded, and the best strategy is selected.

In SalesAction, the Sales Director and Sales Specialist participate. The Sales Specialist creates compelling sales content based on the campaign goal and initial strategies and then discusses it with the Sales Director. The discussion concludes after two content iterations.

In Market_to_sales, the Marketing Director and Sales Director participate. The Sales Director executes the Marketing Director's appealing sales material. The discussion concludes after two execution iterations.

After defining the sales activities in the project, a chain-like workflow is defined and stored in the ChatChainConfig.json file. This workflow defines and controls the project's progress, including specific sales activities in each step and whether data reflection is needed. IdeaGeneration involves the CEO and Marketing Director, focusing on generating creative marketing campaign ideas, requiring reflection but only repeating ten times. Next is ContentGeneration, involving the Marketing Director and Marketing Specialist. Then comes SalesPlanGeneration, where the Sales Director and CEO jointly develop sales strategies to achieve the client's sales objectives. Market_to_sales also takes place, involving collaboration between the Marketing Director and Sales

```
"IdeaGeneration": {
  "assistant_role_name": "Marketing director",
  "user_role_name": "Chief Executive Officer",
  "phase_prompt": [
    "we are super effective at generating viral creative marketing campaign ideas for clients.",
    "Here is the brief from the client:  \"{task}\".",
    "As the {assistant_role}, to make sure we deliver the result for the client, you should keep discussing with me to decide which idea we to deliver that might have the biggest potential and ROI",
    "Note that we must ONLY brainstorm the campaign idea and do not discuss anything else! We should brainstorm and critique on each other's idea, after discussed more than 10 ideas, any of us must actively
  ]
},
"ContentGeneration": {
  "assistant_role_name": "Marketing specialist",
  "user_role_name": "Marketing director",
  "phase_prompt": [
    "According to the goal of the campaign, as well as the initial idea, please generate an ad content for the campaign",
    "Goal: \"{task}\".",
    "Ideas: \"{ideas}\".",
    "As the {assistant_role}, generate a solid social media post for this idea",
    "Note that we must ONLY discuss the content and do not discuss anything else! We should brainstorm and critique on each other's idea, and once we did 2 iterations of content, any of us must actively ter
  ]
```

(a) Action's definition

```
"chain": [
  {
    "phase": "IdeaGeneration",
    "phaseType": "SimplePhase",
    "max_turn_step": -1,
    "need_reflect": "True"
  },
  {
    "phase": "ContentGeneration",
    "phaseType": "SimplePhase",
    "max_turn_step": -1,
    "need_reflect": "True"
  },
  {
    "phase": "SalesPlanGeneration",
    "phaseType": "SimplePhase",
    "max_turn_step": -1,
    "need_reflect": "True"
  },
```

(b) Chat chain

```
"Marketing director": [
  "{chatdev_prompt}",
  "You are Marketing director. Now, we share a common interest in collaborating to successfully complete a task assigned by a new customer.",
  "You direct ChatDev's marketing & creative ad campaign, develop world class marketing campaign idea that define client's brand.",
  "Here is a new customer's task: {task}.",
  "To complete the task, you must write a response that appropriately solves the requested instruction based on your expertise and client"
],
"Marketing specialist": [
  "{chatdev_prompt}",
  "You are Marketing specialist. Now, we share a common interest in collaborating to successfully complete a task assigned by a new customer.",
  "Your main responsibility is writing world class content that resonate with people and reflect client's brand and marketing campaign.",
  "Here is a new customer's task: {task}.",
  "To complete the task, I will give you one or more instructions, and you must generate one of the best content you've ever written in your life."
],
```

(c) Role's definition

Fig. 8. Multi-agent environment setup

Director to convert marketing strategies into sales actions. Finally, SalesAction, involving the Sales Director and Sales Specialist, focuses on implementing sales strategies and producing sales materials.

This project successfully developed a software application aimed at boosting the sales of "Natural Language Processing: A Textbook with Python Implementation." Throughout this process, each role made significant contributions: the company team configured the basic information for the project; the prompt engineer improved user query prompts for better utilization of large language models; the user specified the requirements for software redesign; the marketing director and CEO jointly determined the marketing strategy; the sales director and CEO discussed and decided on the sales strategy; the sales expert was responsible for creating the sales and promotional content. Ultimately, the team collectively developed a mobile application that effectively promoted the book by showcasing its key features, recommendations from industry experts, and a user-friendly interface, thereby enhancing sales outcomes.

IV. Conclusion

In conclusion, this project addresses the long-standing quest for artificial intelligence at or above human level by focusing on the development of AI agents. AI agents, as artificial entities capable of perceiving the environment, making decisions, and taking actions, hold great promise in achieving this goal. While previous efforts have primarily concentrated on algorithmic advancements and training strategies for specific tasks, a notable gap in the field is the absence of a universal and robust model that can adapt to various scenarios.

Large language models (LLMs) have emerged as a potential catalyst for the pursuit of Artificial General Intelligence (AGI) due to their versatility. Researchers have leveraged LLMs as a foundation for building AI agents and have made substantial progress in this direction.

The practical implementation of this project involved rebuilding a GPT3.5-based multi-agent environment, simulating a marketing company. Six distinct roles were defined, each with specific responsibilities, including the CEO, Counselor, Marketing Director, Marketing Specialist, Sales Director, and Sales Specialist. These roles collaborated within a structured workflow involving Idea Generation, Content Generation, Sales Plan Generation, Sales Action, and Market-to-Sales phases.

The project's plan to develop an LLM-based agent in the marketing industry exemplifies the potential of this approach to optimize sales strategies. Through various phases, the agents generated creative marketing ideas, developed content, devised sales plans, executed sales actions, and converted marketing strategies into sales activities. This practical experiment aimed to achieve the optimal sales plan within a simulated marketing company environment.

By bridging the historical evolution of artificial intelligence, the philosophical underpinnings of agency, and the contemporary capabilities of LLMs, this work contributes to the ongoing discourse on creating adaptive and universal AI agents capable of pushing the boundaries of artificial intelligence.

References

[1] S. J. Russell, "Artificial intelligence a modern approach," Pearson Education, Inc., 2010.

[2] D. Diderot, "Diderot's early philosophical works," Open Court, 4th ed., 1911.

[3] A. M. Turing, "Computing machinery and intelligence," Springer, 2009.
[4] M. J. Wooldridge and N. R. Jennings, "Intelligent agents: theory and practice," Knowl. Eng. Rev., vol. 10, no. 2, pp. 115–152, 1995.
[5] M. Schlosser, "Agency," in The Stanford Encyclopedia of Philosophy, E. N. Zalta, Ed., Metaphysics Research Lab, Stanford University, Winter 2019 edn., 2019.
[6] G. A. Agha, "Actors: a Model of Concurrent Computation in Distributed Systems (Parallel Processing, Semantics, Open, Programming Languages, Artificial Intelligence)," Ph.D. dissertation, Univ. Michigan, USA, 1985.
[7] S. Green, L. Hurst, B. Nangle, et al., "Software agents: A review," Dept. Computer Science, Trinity College Dublin, Tech. Rep. TCS-CS-1997-06, 1997.
[8] M. R. Genesereth and S. P. Ketchpel, "Software agents," Commun. ACM, vol. 37, no. 7, pp. 48–53, 1994.
[9] R. Goodwin, "Formalizing properties of agents," J. Log. Comput., vol. 5, no. 6, pp. 763–781, 1995.
[10] J. S. Park, J. C. O'Brien, C. J. Cai, et al., "Generative agents: Interactive simulacra of human behavior," CoRR, abs/2304.03442, 2023.
[11] Y. Shoham, "Agent oriented programming," in Knowledge Representation and Reasoning Under Uncertainty, Logic at Work, M. Masuch and L. Pólos, Eds., Lecture Notes in Computer Science, vol. 808, Springer, pp. 123–129, 1992.
[12] R. A. Brooks, "Intelligence without representation," Artificial intelligence, vol. 47, no. 1-3, pp. 139–159, 1991.
[13] P. Maes, Designing autonomous agents: Theory and practice from biology to engineering and back, MIT press, 1990.
[14] R. A. Brooks, "A robust layered control system for a mobile robot," IEEE J. Robotics Autom., vol. 2, no. 1, pp. 14–23, 1986.
[15] R. A. Brooks, "Intelligence without reason," in The artificial life route to artificial intelligence, Routledge, pp. 25–81, 2018.
[16] M. Hutter, "Universal Artificial Intelligence: Sequential Decisions Based on Algorithmic Probability," Springer Science & Business Media, 2004.
[17] D. K. B. Seng and C. H. Tan, "Artificial Intelligence and Agents," NUS Centre for Technology, Robotics, Artificial Intelligence & the Law Working Paper 21/02, to be published in E. Lim and P. Morgan (eds), The Cambridge Handbook of Private Law and Artificial Intelligence (Cambridge Univ. Press, Forthcoming), NUS Law Working Paper No. 2021/019, City Univ. Hong Kong School of Law Legal Studies Research Paper No. Forthcoming, 2022.
[18] T. Yigitcanlar, K. Degirmenci, and T. Inkinen, "Drivers behind the public perception of artificial intelligence: insights from major Australian cities," AI & Soc, 2022.
[19] Y. Roh, G. Heo, and S.E. Whang, "A Survey on Data Collection for Machine Learning: A Big Data - AI Integration Perspective," J. Machine Learning, vol. 3, no. 3, pp. 21-30, 2022.
[20] S. Chowdhury, O. Wigertz, and B. Sundgren, "Artificial Intelligence Methods in Data Analysis and Interpretation," J. Data Analysis, vol. 6, no. 6, pp. 51-60, 2025.
[21] P. Wang, C. Hahm, and P. Hammer, "A Model of Unified Perception and Cognition," J. Artificial Intelligence, vol. 1, no. 1, pp. 1-10, 2020.
[22] E. M. Ditria, C. A. Buelow, M. Gonzalez-Rivero, and R. M. Connolly, "Artificial intelligence and automated monitoring for assisting conservation of marine ecosystems: A perspective," Front. Marine Science, vol. 9, 2022, DOI: 10.3389/fmars.2022.918104.
[23] A. S. Khritankov, "Understanding Real-World AI Planning Domains: A Conceptual Framework," J. Artificial Intelligence, vol. 1, no. 1, pp. 1-10, 2023.
[24] C. Andrews, K. Cooke, A. Gomez, P. Hurtado, T. Sanchez, S. Shah, and N. Wright, "AI in Planning: Opportunities and Challenges and How to Prepare," American Planning Association, Sept. 9, 2022.
[25] M. Shvo, S. Sohrabi, and S. A. McIlraith, "An AI Planning-Based Approach to the Multi-Agent Plan Recognition Problem," in Canadian AI 2018: Advances in Artificial Intelligence, pp. 253–258, 2018.
[26] A. S. Khritankov, "Analysis of hidden feedback loops in continuous machine learning systems," arXiv preprint arXiv:2101.05673, 2021.
[27] C. Qian, X. Cong, C. Yang, W. Chen, Y. Su, J. Xu, et al., "Communicative agents for software development," arXiv preprint arXiv:2307.07924, 2023.
[28] S. Hong, X. Zheng, J. Chen, Y. Cheng, C. Zhang, Z. Wang, et al., "Metagpt: Meta programming for multi-agent collaborative framework," arXiv preprint arXiv:2308.00352, 2023.
[29] W. Chen, Y. Su, J. Zuo, C. Yang, C. Yuan, C. Qian, et al., "Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors in agents," arXiv preprint arXiv:2308.10848, 2023.
[30] J. Lin, D. Fried, D. Klein, et al., "Inferring rewards from language in context," in Proc. of the 60th Annual Meeting of the Association for Computational Linguistics (Vol. 1: Long Papers), Dublin, Ireland, May 22-27, 2022, pp. 8546-8560.
[31] J. Wei et al., "Finetuned Language Models Are Zero-Shot Learners," Learning, Sep. 2021.
[32] K. Valmeekam, A. Olmo, S. Sreedharan, and S. Kambhampati, "Large Language Models Still Can't Plan (A Benchmark for LLMs on Planning and Reasoning about Change)," arXiv preprint arXiv:2206.10498, 2022.
[33] K. Valmeekam, S. Sreedharan, M. Marquez, A. Olmo Hernandez, and S. Kambhampati, "On the Planning Abilities of Large Language Models: A Critical Investigation with a Proposed Benchmark," arXiv preprint arXiv:2302.06706, 2023.
[34] J. Code, "Agency for Learning: Intention, Motivation, Self-Efficacy and Self-Regulation," Front. Education, vol. 5, 2020.
[35] D. Driess et al., "PaLM-E: An Embodied Multimodal Language Model," Mar. 2023.
[36] Y. Pan, T. Yao, Y. Li, et al., "X-linear attention networks for image captioning," in 2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition, Seattle, WA, USA, June 13-19, 2020, pp. 10968-10977.
[37] M. Cornia, M. Stefanini, L. Baraldi, et al., "M2: Meshed-memory transformer for image captioning," CoRR, vol. abs/1912.08226, 2019.

[38] J. Lin, Y. Du, O. Watkins, et al., "Learning to model the world with language," CoRR, vol. abs/2308.01399, 2023.
[39] "AI in Ecommerce: Applications, Benefits, and Challenges," Hostinger, Nov 15, 2023.
[40] M. Mirbabaie, S. Stieglitz, and N. R. J. Frick, “Artificial intelligence in disease diagnostics: A critical review and classification on the current state of research guiding future direction,” Health Technol., vol. 11, pp. 693–731, 2021.
[41] E. B. Boukherouaa, K. AlAjmi, J. Deodoro, A. Farias, and R. Ravikumar, “Powering the Digital Economy: Opportunities and Risks of Artificial Intelligence in Finance,” Dep. Pap., Int. Monet. Fund, Oct. 22, 2021.
[42] F. Cugurullo, “Urban Artificial Intelligence: From Automation to Autonomy in the Smart City,” Front. Sustain. Cities, vol. 2, Jul. 30, 2020.
[43] T. F. Blauth, O. J. Gstrein, and A. Zwitter, “Artificial Intelligence Crime: An Overview of Malicious Use and Abuse of AI,” 2022.
[44] “India vs Australia 3rd ODI, Highlights: Australia win series 2-1, become No.1,” The Indian Express, Mar. 22, 2023.