Optimization of Decision Making Capabilities of Intelligent Characters in Strategy Games Based on Large Language Model Agent

Kehua Shi
School of Fuzhou University, Fujian, China

Abstract: In order to explore the optimization of decision-making ability of intelligent characters in strategy games based on Large Language Model (LLM) Agent, this paper takes pokellmon[1] developed by Georgia Institute of Technology as an example, and realizes the exploration of the scheme of optimization of decision-making ability by imitating the human players: 1. Collecting the experience of various human players and letting the LLM to expand the knowledge by means of Prompt Engineering and RAG(Retrieval-Augmented Generation) to get that experience; 2. Collecting the records of human high-scoring matchups records from the The replays records on pokemon showdown official website, and make a dataset of multiple rounds records of each game in the mode of multi-round dialogues, and fine-tune the LLM through QLoRA fine-tuning method (Tim Dettmers et al., 2023) after processing the data, so as to allow the LLM to learn the decision-making of the human masters in the random battles. The open-source LLM llamma3.1-8b was chosen for this study, and the battles against the heuristic bots showed that the modified pokellmon got some degree of strategy optimization, and compared to the very beginning of the battles against the heuristic bots, the winning rate was improved from 18% to 34%.

Keywords: LLM Agent; Prompt Engineering; RAG; QLoRA; Game Strategе

1. Introduction

Strategy games are a complex and diverse game genre that require players to think deeply about resource management, long-term planning, and decision making. Due to its complexity and diversity, strategy games are well suited for research in the field of artificial intelligence. Meanwhile, with the continuous and deep development of the game field, players' requirements for intelligent roles in the game are getting richer and richer, and more and more different technologies are applied to intelligent roles in strategy games. For the field of artificial intelligence, exploring theoretical approaches to improve the strategic planning ability of intelligences helps intelligences to solve more complex tasks. For games, AI roles make the game human-machine behavior uncertain, which can make the game more diversified and keep the player's freshness to a certain extent; while more intelligent human-machine improves the difficulty of the game, which makes the game more challenging. In addition, it can promote the application of the AI in the field of gaming, and raise people's AI attention.

2. Literature Review

Common AI techniques in games are traditional pathfinding algorithms, such as the A* search algorithm for heuristic pathfinding and improved algorithms based on the A* search algorithm; Finite-state machine and Behavior tree are also widely used in game AI, as well as HTN (Hierarchical Task Network). However, the above algorithms are written by human beings, and players can quickly discover the laws of them after certain observations, thus affecting the game experience. And GOAP (Goal Oriented Action Planning) algorithm can autonomously and rationally select relevant actions to accomplish the decision-making goal[5]. However, its ability to respond to environmental changes in real time is limited and requires predefined goals and actions.

Reinforcement learning appeared earlier as a technical approach and has had applications in gaming since the 1990s (e.g., TD-Gammon). Although the concept of neural networks was introduced earlier, modern deep learning applications (e.g., CNNs) began to be widely

used in fields such as image processing only after 2006, and gradually extended to the gaming field. However, compared to traditional decision-making algorithms, the training process of deep learning is not convenient for debugging, and the effect is more uncontrollable[5]. Deep reinforcement learning (DRL), which began to emerge in 2013, combines deep learning and reinforcement learning. It learns by interacting with the environment and optimizes based on feedback signals, which significantly improves AI performance in games and is well suited to dynamically changing game environments that require real-time strategy adjustments. In 2016, DeepMind's AlphaGo defeated the world champion in Go by combining deep learning and Monte Carlo Tree Search (MCTS). This event became a milestone for AI in strategy games; 2019's AlphaStar in StarCraft II demonstrated the potential of DRL in complex real-time strategy games. However, it relies on environment modeling, which requires high state space design, and will face the trade-off between exploration and exploitation. In addition to this, DRL requires a large amount of training data and time, which is less efficient.

With the rapid development of large language model technology, the application of LLM Agent in the game field has attracted widespread attention, and many related applications have appeared one after another, such as Stanford Town and the application of LLM Agent in the game Minecraft, Voyager, etc. LLM Agent has demonstrated powerful reasoning and decision-making capabilities by processing and generating natural language, and has become an important tool to enhance the game experience and intelligence level. It can generate coherent strategies based on historical information over multiple rounds, which is suitable for long term planning, and unlike Deep Reinforcement Learning, LLM does not require a lot of training ahead of time to make basic decisions. However, the performance of LLM Agent in strategy games still has a lot of room for improvement, and there are still some problems, such as:1. Single strategy: LLM Agent may reuse known strategies and lack of innovation and diversity.2. Strategy stability: Under different game environments and opponents' strategies, it may be difficult for LLM Agent to maintain stable strategy performance.3. Real-Time Feedback Challenge: Adaptability to real-time feedback may be lacking in the decision-making process of LLM Agent

3. Method

The purpose of this study is to explore the decision optimization of intelligent characters based on LLM Agent, therefore, the focus of this study is on decision optimization, i.e., to find the optimization method of LLM's decision making in the game, rather than optimizing the performance data of LLM in the game.

To improve the performance of LLM Agent-based game AI characters in strategy games, the first and foremost thing is to improve the LLM's strategic ability against the game. The advantage of intelligent characters over human players is the ability to clearly record the complex rules and various data of the game, which is often difficult for human players to memorize all of them. The advantage of human players over intelligent characters is that human players are able to summarize their experience based on previous game battles, and make more complex and long-term strategies, and even make “deceptive” psychological tactics for the game. Therefore, this study aims to improve the performance of LLMs in strategy games by allowing them to mimic humans and learn from their experience.

This decision optimization study takes the pokellmom developed by the team as an example, which is a LLM Agent based on the pokemon showdown game, and plays as a player in the game. Based on cost constraints, the open-source model llama3.1-8b was chosen as the agent's “brain”in this study to make decisions in the game, and the game battles were mainly based on the gen8 random battle mode of pokemon showdown.

3.1 Experience Learning

In this study, we collected and organized the pokemon showdown battle experience of different players from reddit website, smogon website, quora website, etc, including macro and general experiences, such as “Defense strategies”, which is designed to protect your Pokemon and keep them alive as long as possible. This can include using defensive abilities to increase your Pokemon’s resistance or using healing abilities to heal them after taking damage.Advantages of this strategy: They allow you to last longer in a match, buy time, and give you the ability to counterattack. Disadvantages: it can slow down the pace of the

match and allow the opponent to gain an advantage.", and micro and specific experience including recommended combos for pokemon moves, items, etc..

3.1.1 RAG

In order to let LLM have the collected experience knowledge, this study builds a RAG() experience knowledge base, which facilitates LLM to retrieve the information related to the battle data in the knowledge base according to the battle situation on the field when making decisions, and measures and combines the retrieved experience with the original decision-making method of Pokellmon, so that the LLM can finally choose the appropriate action. However, there are some experience in the knowledge base that are related to specific pokémon, pokémon moves, items, weather and other combination conditions, therefore, after retrieving the experience information from the knowledge base, it is necessary to choose according to the specific status quo of the game battle, for example, the experience suggests that certain moves and items should be used in combination, but the pokémon player may not have all the moves and items, therefore, the LLM's decision-making method will be based on the information in the knowledge base.

In this study, the collected experience information is firstly organized into a document, then the document is segmented, and the segmented data is vectorized and stored in a vector database. When the game requires pokellmon to make a decision, the data of the round, such as the number of pokellmon available to it's own side, what is the state of each pokellmon, what are the moves, what items can be used, how are the weather conditions on the field, and how many pokellmon are available to the opponent, what are the known opponent pokellmon, etc., are retrieved in the vector database, and the results will be added into the pre-decision prompt submitted to the LLM, which is required to make the decision for the next move based on the current state of the game and the relevant empirical knowledge.

3.1.2 Prompt Engineering

However, while specific experience information is easy to retrieve, macroscopic and general experience information is not easy to retrieve, because there are few specific pokemon names, actions, items, etc. in these macroscopic experience. Therefore, in order to utilize this macro-experience information, it is added to the system prompt of LLM, which is used to prompt LLM's decision-making skills and choices.

Table 1. Performance of Experience Learning in Battles Against the Bot

Player	Win rate ↑	Score ↑	Turn #	Battle #
LLaMA-3.1	18.00%	4.33	19.01	100
Experience	25.00%	4.51	20.27	100

3.1.3 Results

Human players have accumulated a great deal of experience in the process of game battles, which contains the understanding of tactics, familiarity with different attributes and pokemon, and prediction of opponent's behavior. By transforming these effective tactics into prompts, LLM Agent with strong language ability can quickly absorb and apply this valuable knowledge, understand the advantages and disadvantages of various tactics, and avoid inefficient exploration to a certain extent. Not only that, human players can recognize specific battle situations and adjust their strategies based on their experience. By providing this experience, the LLM Agent is able to acquire the knowledge to recognize common battle patterns and generate strategies to improve the win rate when facing similar situations. In addition, the provision of multiple human player strategies enhances the tactical flexibility of the LLM Agent, allowing it to dynamically adjust to the opponent and the battle environment, and to a certain extent reduces the homogeneity of strategies.Rapid Attack, Defense, and Counter Attack strategies are the most common and effective, utilizing different core gameplay mechanics and adapting to most matchmaking scenarios, while also being easier to master and implement. Rapid Attacks are complemented by high bursts to overwhelm opponents, Defense Strategy counters quick attacks through endurance and attrition., and Counter Attacks take advantage of opponents' mistakes in the battlefield.

Taking llama 3.1-8b as the core of the agent's thinking, the performance of the original pokellmon and the pokellmon with the modified prompt and the added experience knowledge base in the battle against the robots is shown in Table 1, which shows that the improved agent has a better performance in decision making, and it can be reasonably hypothesized that if more and higher-quality experience data is collected, it can make LLM's decision-making in the game further optimized.

3.1.4 Advanced strategies

In addition to the common basic strategies, some human players use more advanced strategies, such as deception strategies. A deception strategy involves misleading an opponent through false signals or actions, inducing them to make a wrong judgment or take an unfavorable action, and then using these mistakes of the opponent to gain an advantage. This can be accomplished in a variety of ways, including bluffing, disguising one's strategy, feigning weakness and thereby inducing the opponent to switch, or incorrectly anticipating the situation.

The information provided to the LLM Agent about deception strategies as a prompt includes a description of the deception strategy, specific practices, advantages and disadvantages of the strategy, and is intended to rely on the powerful language capabilities of LLM to understand what a deception strategy is, to understand the high rewards and risks of deception strategies, to make decisions on its own based on the historical record and the current situation, and to be flexible in evaluating or adjusting the strategy.

However, there are still challenges and limitations for LLM Agent to execute advanced strategies.The core ability of LLM is based on language, which lacks the understanding and speculation of the opponent's psychological state, and relies more on the objective data of the opponent, making it more difficult to grasp the deep psychological aspects (e.g., dealing with the opponent's emotions, stress, etc.), and it is difficult to predict the opponent's emotional fluctuations and changes in mindset as human players do, and to determine when it is appropriate to execute psychological strategies.

It is evident that while LLM has the potential to execute advanced strategies, further exploration is needed to master and execute these complex advanced strategies well.

3.2 Replays Learning

In addition to the advantage of game experience over intelligent characters, human beings can learn from watching exciting and high-quality battles, thus realizing leveling up and improving their performance in the game. Therefore, this study collects 100 gen8 random battle replays from the official pokemon showdown website from high to low rating as a dataset, and learns by means of large model fine-tuning.

3.2.1 Dataset

What LLM needs to learn is to make decisions based on the current game situation, i.e., input the situation information and output the player decisions. However, because the game is turn-based, and a round of the game contains multiple matchmaking rounds, if only the decision of the last round is used as the output data, there will be a serious loss of information[4], if the multiple rounds are split into multiple samples, and each sample input contains all the previous input data, then the model will not be able to learn the overall game information and there is a large number of repetitive operations. Therefore, in order for the LLM to learn the information of the whole game, this study adopts a large model fine-tuning similar to the multi-round dialogues, where each round is a round of dialogues, and directly constructs outputs that include all the decisions taken by the LLM in multiple rounds, which makes full use of all the information of the decisions, and at the same time does not have to split and repeat the computation, which is very efficient.

The sample inputs to the training dataset are multiple rounds of the user's prompt, each of which includes a record of the battle at the most recent round (or the very beginning of the game) (including information such as the appearing pokemon, the move used, the pokemon's STATUS, the pokemon's attack, and damage), the system prompt to the LLM, and the curent battle state (describing the current pokemon status of the side and the opponent and their possible moves for LLMs decision making), and the output of the dataset is the actions taken by own side in each round, including switch and move. since LLM needs to learn the behavior of the winning side, the dataset is constructed with the winner as the side and the loser as the other side.

In order to obtain the game battle information, this dataset mainly analyzes the “battle-log-data” information in the replays html file line by line, and updates the state of the player object and pokemon object according to the obtained game process information, which can be used as the current battle state.

Table 2. Performance of Fine-Tuning in Battles Against the Bot

Player	Win rate ↑	Score ↑	Turn #	Battle #
Origin	25.00%	4.51	20.27	100
Fine-tuning	34.00%	5.22	20.18	100

For example, if we get the “switch” field, we

analyze this line to get the player and pokemon information, and set the active pokemon. However, since there is no player's perspective in the replay log, we don't know in advance which Pokémon are available to the player and which moves are available to the pokémon, so we need to update the information according to the battle-log-data. In addition to this, there are fields such as “move”, “-status”, and “-damage”.

3.2.2 Fine-tuning

After acquiring the dataset, this study employs the QLoRA[2] method to fine-tune the LLM so that it learns the decision-making behavior of each game in the dataset. Where QLoRA is an effective fine-tuning fine-tuning method for quantitative large models, which centers on maintaining or even improving the model's performance on a specific task while significantly reducing the graphics memory requirement through innovative quantization methods and memory management techniques.

3.2.3 Results

The performance of pokellmon using the fine-tuned LLM in the battle against the robot is shown in Table 2, which shows that the LLM fine-tuned has better performance in game decision making, and subsequently can try to get better performance by increasing the number of samples in the training set.

4.Conclusion

Based on pokellmon, this work explores a decision optimization scheme for an intelligent character based on LLM Agent, including collecting human players' experience, expanding the knowledge of LLM by means of prompt engineering and RAG, as well as collecting records of human high-scoring matchups and making a dataset of the round records of each game by means of a multi-round dialogue mode, and fine-tuning LLM by means of QLoRA method to fine-tune the LLM so that the LLM learns the decisions of human masters in random battles. The strategy method is generalized and can be used for strategy enhancement of LLM Agent-based intelligent characters in other games.

Battles against heuristic bots show that Modified pokellmon based on llama3.1-8b can get a certain degree of strategy optimization, and compared to the very beginning of the battle against the heuristic bots, the win rate is improved from 18% to 34%.

In addition, since this study focuses on exploring the strategy enhancement options and feasibility of LLM rather than optimizing the performance to the extreme, the experimental results of this study using LLAMA 3.1-8b are not as good as the original pokellmon using GPT4, but there is still some room for improvement in this study's method, such as collecting more and higher quality human player experience to expand the knowledge of LLM, as well as collecting more replays recordings of human masters' game matches and producing them as sample datasets for fine-tuning.

References

[1] Hu, Sihao, Tiansheng Huang, and Ling Liu. “Pok\'eLLMon: A Human-Parity Agent for Pok\'emon Battles with Large Language Models.” arXiv preprint arXiv:2402.01118 (2024).

[2] Dettmers, Tim, et al. “Qlora: Efficient finetuning of quantized llms.” Advances in Neural Information Processing Systems 36 (2024).

[3] Ma, Weiyu, et al. “Large language models play starcraft ii: Benchmarks and a chain of summarization approach.” arXiv preprint arXiv:2312.11865 (2023).

[4] CSDN. (2024) BaiChuan13B Examples of fine-tuning multiple rounds of dialogues https://blog.csdn.net/Python_Ai_Road/article/details/132400115?rId=132400115&source=Freyr_s &type = blog&refer = APP

[5] cnblogs. (2023) Game AI Behavioral Decision Making https://www.cnblogs.com/OwlCat/p/17871494.html

