

Title	大規模言語モデルを報酬計算に用いたキャラクターらしいNPCの強化学習
Author(s)	徳永, 遼太
Citation	
Issue Date	2025-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/19840
Rights	
Description	Supervisor: 池田 心, 先端科学技術研究科, 修士 (情報科学)

修士論文

大規模言語モデルを報酬計算に用いたキャラクターらしいNPCの強化学習

徳永 遼太

主指導教員 池田 心

北陸先端科学技術大学院大学
先端科学技術専攻
(情報科学)

令和7年3月

Abstract

In recent years, artificial intelligence (AI) technology has rapidly advanced, leading to extensive research and applications across various fields, including games. In particular, large language models (LLMs) such as ChatGPT have gained significant attention because of their high capabilities and broad applicability.

One example of AI technology applied to games is the behavior control of non-player characters (NPCs), which are not operated by human players. NPCs serve as enemies, teammates, or inhabitants of the game world, playing a crucial role in enhancing the overall gaming experience. Traditionally, they have been implemented primarily using rule-based methods designed by developers. However, in recent years, AI techniques such as supervised learning, reinforcement learning, and tree search have been actively adopted for NPC behavior control.

In the era when AI technology was still immature, most complaints about NPC behaviors stemmed from issues such as “not acting appropriately, being too weak.” However, as AI players have surpassed human skill levels in many games, including Go, Mahjong, and StarCraft, a new set of challenges has emerged. AI-controlled NPCs can sometimes be overly strong or exhibit behavior that appears unnatural to human players, which may negatively impact the gaming experience. As a result, research has increasingly focused on developing NPCs that prioritize human-like behavior over skill levels.

For example, Maia is a model trained through supervised learning on a large dataset of amateur chess games. It has been shown to predict human moves more accurately than players developed using tree search or reinforcement learning. In another approach, Fujii et al. created a human-like Super Mario player by incorporating biological constraints, such as “cognitive fluctuations,” “delays between cognition and action,” and “operation fatigue,” into reinforcement learning agents.

In story-driven games with unique world settings, such as role-playing games (RPGs), NPCs are expected to behave not only like generic humans but also in a manner that aligns with their specific character roles. Even among warriors with the same abilities, there can be various characterizations, such as a timid warrior, a brave warrior, an attention-seeking warrior, or a warrior who secretly wishes for their teammates to die. Each of these character roles requires different behaviors.

Modern games feature a vast number of NPCs, and their appropriate behaviors are required not only in predefined maps and events but also in randomly generated situations. In such cases, implementing character behaviors using rule-based systems is impractical, and the diversity of situations makes it difficult to collect a large amount of training data. To address this, reinforcement learning has been explored as an approach where NPCs receive higher rewards for behaving in a manner consistent with their character. However, designers still need to define

“what kind of behavior is considered character-like,” and as the complexity of the problem increases, this becomes increasingly challenging.

In this work, we hypothesized that LLMs have the ability to “understand a given world setting and situation and reasonably judge desirable states and actions.” Based on this assumption, we explored the idea of leveraging LLMs for reward calculation in reinforcement learning. Roughly speaking, we propose a framework where the LLM is given an instruction such as: “In this world setting, the character has the following roles. The current situation is as follows. The agent has taken this sequence of actions, leading to this outcome. Evaluate whether this action sequence is appropriate for the character, providing a score with reasoning.” The obtained evaluation score is then used as an episodic reward in reinforcement learning to guide the NPC’s behavior.

The first experiment involved a scenario where a royal guard needed to reach a destination without crossing in front of the hero, who was having an audience with the king. In many cases, the agents successfully learned the necessary and sufficient detour route we had intended, and the reasoning provided by the LLM for its evaluations aligned with our expectations. However, in some trials, the LLM highly rated routes where the guard wandered unnecessarily around the room, leading to the unintended learning of such behavior. We consider this an unintended side effect of our approach where episodes with high evaluations were used as samples to further refine the LLM’s evaluations.

The second experiment involved a scenario where a party consisting of a hero, a princess, and a cleric engaged in battle with a slime. The task was to train the cleric, who could perform both attacks and healing actions, to behave according to different character roles. We tested three distinct role settings: (1) prioritizing the princess’s safety above all else, (2) being aggressive and enjoying battles, and (3) being extremely cautious and timid. As a result, the cleric exhibited behaviors appropriate to each role settings: sacrificing their own well-being to protect the princess, attacking the slime even when the hero or princess was injured, and exclusively focusing on healing.

Through the experiments in this work, we observed that LLMs have a certain ability to “reasonably judge desirable states and actions.” However, we also found that LLMs’ judgments can vary significantly depending on how the instructions are phrased. Future research will need to explore methods to improve consistency in these evaluations.

概要

近年、人工知能（AI）技術は急速に発展し、ゲームを含むさまざまな分野で研究と活用が進んでいる。特に、ChatGPTをはじめとする大規模言語モデル（LLM）は高度でまた応用範囲が広く、注目されている。

ゲームへのAI技術の活用の一例として、人間プレイヤーが操作しないキャラクタ（NPC）の行動制御が挙げられる。NPCは敵、味方、ゲーム世界の住人として機能し、ゲームの面白さを支える重要な要素の一つである。古くは主に開発者が設計したルールベース方式で実装されてきたが、近年では教師あり学習、強化学習、木探索といったAI技術の導入も活発に行われている。

AI技術が未熟な時代には、NPCの挙動に対する不満の多くは「適切な行動をしない、弱い」という問題に起因していた。しかし、囲碁や麻雀、StarCraftなど多くのゲームでNPC（AIプレイヤー）の技量が人間を上回るに至ると、また別の問題が注目されるようになった。AIによるNPCは、強すぎたり、人間からすると不自然に見える行動を取ったりすることがあり、これがプレイヤーのゲーム体験を損ねることがある。そこで、強さよりも、人間らしさを重視したNPCの作成方法が重点的に研究されるようになってきている。

例えばMaiaはチェスのアマチュアプレイヤーの棋譜を大量に集めて教師あり学習を行ったモデルであり、これが木探索や強化学習で作ったプレイヤーよりも人間の着手を予想しやすいことが示された。あるいは、藤井らは、人間の持つ「認知のゆらぎ」「認知から行動までの遅れ」「操作疲れ」といった生物学的な制約を強化学習エージェントに導入することで、人間らしく見えるスーパーマリオのプレイヤーを作成している。

ロールプレイングゲーム（RPG）など独自の世界観を持つストーリー性の高いゲームでは、各NPCが単に一般の人間らしくふるまうだけでなく、“そのキャラクタらしく”ふるまうことが求められる。同じ能力を持った戦士であっても、臆病な戦士、勇敢な戦士、目立ちたがりの戦士、実は仲間に死んで欲しいと思っている戦士など、さまざまなキャラクタ付けが行われている場合があり、それぞれに求められる行動は異なる。

昨今のゲームには非常に多くのNPCが登場し、既定のマップやイベントのみならず、ランダムに生成された状況においても適切なNPCの振る舞いが求められることもある。このような場合に、キャラクタの行動をルールベースで実装することは非現実的であり、また状況がさまざまであることから、訓練データを大量に集めることも難しい。そこで、強化学習を用いて、キャラクタらしい振る舞いを行った場合に高い報酬を与えるようなアプローチが試みられている。しかし、結局「どんな振る舞いをしたらキャラクタらしいのか」はデザイナーが設定してやる必要があり、問題が複雑になるほどそれは簡単ではないことが分かっている。

そこで本研究では、LLMが「与えられた世界観や状況を理解し、常識的に好ましい状態や行動を判断する能力」を持っていると仮定し、これを強化学習の報酬の計算に援用するという着想に至った。簡単にいうと、LLMに「こういう世界観

で、キャラクターはこういう設定です。こういう状況です。エージェントはこういう行動系列をとって、結果的にこうなりました。この行動系列は、このキャラクターにふさわしいですか？理由付きで点数で評価してください」というような指示を出し、得られた評価値を、強化学習のエピソード単位の報酬として与えて学習させるという枠組みを提案した。

実験1は、勇者が王様に謁見しているシーンで、近衛兵がそれを横切らずに目的地まで適切なルートで移動することができるかという課題を設定した。我々が想定していた必要十分な迂回ルートを学習してくれることもあり、その際の評価理由は期待した通りであった。一方で、試行によっては、謁見室内を無駄に歩き回るようなルートがLLMによって高く評価され、それが学習されてしまうこともあった。これは、高い評価を得たエピソードをサンプルとしてLLMに与えるという我々の工夫が起こした副作用であると考えている。

実験2は、勇者と姫と僧侶のパーティが、スライムと戦うというシーンで、攻撃と回復を行える僧侶の行動を学習する課題を設定した。僧侶には、「姫の身の安全を最優先する」「血の気が多くて戦いを好む」「非常に慎重かつ臆病な性格」という3つの異なるキャラクター付けを試した。結果として、それぞれ、自分が傷ついても姫を助ける、姫や勇者が傷ついてもスライムを攻撃する、および回復しかしない、といった、キャラクター付けにふさわしい挙動を確認することができた。

本研究の実験を通して、LLMには一定の「常識的に好ましい状態や行動を判断する能力」が認められたが、指示の仕方一つで何を好ましいとするかは大きく変わることが確認された。これを改善するための工夫が今後は必要になると考える。

目次

第1章	はじめに	1
第2章	関連研究	3
2.1	ゲームにおけるキャラクターらしさ	3
2.2	LLMのゲームへの応用事例	4
第3章	提案手法	5
第4章	実験1：謁見の間にふさわしい移動経路	8
4.1	対象ゲーム	8
4.2	学習方法	10
4.3	報酬計算	10
4.4	学習結果	11
4.5	与える指示の影響	13
第5章	実験2：キャラクターの背景設定による戦闘シーンの挙動変化	16
5.1	対象ゲーム	16
5.2	キャラクターの背景設定	17
5.3	学習方法と報酬計算	17
5.4	学習結果	19
第6章	おわりに	22

目 次

4.1	対象ゲームのイメージ	9
4.2	最短経路の例	9
4.3	迂回経路の例	9
4.4	試行1での最高評価の経路	12
4.5	試行2での最高評価の経路	12

表 目 次

5.1 僧侶の背景設定「姫の身の安全を最優先する.」での対照的な評価の比較	19
5.2 僧侶の背景設定「血の気が多く戦いを好む.」での対照的な評価の比較	20
5.3 僧侶の背景設定「非常に臆病かつ慎重な性格.」での対照的な評価の比較	21

第1章 はじめに

近年、人工知能（AI）技術は急速に発展し、多様な分野で研究と応用が進んでいる。特に、2022年に登場したChatGPTをはじめとする大規模言語モデル（LLM）は、社会の多くの領域での活用や導入が進んでいる。

AI技術はビデオゲーム分野と密接な関りを持つ。ビデオゲームは、新しく開発されたAI技術の性能を評価するテストベッドとして用いられることも多い。これは、ビデオゲームはロボティクスなどの現実世界での活用と比べ「状態の観測」「大量の試行」「評価指標の定義」が比較的容易であるためAI技術との親和性が高いことなどの理由からであると考えられる。

ゲームにおけるAI技術の活用の一例として、ノンプレイヤーキャラクタ（NPC）の行動制御が挙げられる。NPCはゲームにおいて擬似的な対戦相手やゲーム内世界の住人として機能し、ゲームの面白さを支える重要な要素の一つである。これまで、NPCの行動制御にはゲーム開発者が設計したルールベース方式が主に用いられてきたが、教師あり学習、強化学習、木探索といったAI技術の導入も活発に研究されている。

従来のNPCの研究では、ゲームにおける「強さ」の向上が主な目標とされてきたが、近年では「人間らしさ」の向上にも注目が集まっている。多くのゲームでAIの強さが人間と同じかそれ以上のレベルに達したこと、人間よりも強すぎるAIはプレイヤーの対戦相手として適さない場合があること、などから「強さ」は実用上十分な水準に達したと捉えられていることが理由の一つである。また、強さを重視して設計されたAIは人間とは異なる非直感的な振る舞いや戦略を取ることがあり、プレイヤーに違和感を与える可能性を抑制するために「人間らしさ」が必要となる。藤井らは、ゲームプレイにおける人間の生物学的制約として「身体的な制約（“ゆらぎ”，“遅れ”，“疲れ”）」や「生存の欲求（“訓練と挑戦のバランス”）」を定義し、これらを強化学習に取り入れることで、自動的に「人間らしい」NPCを構築する手法を提案した [1]。

また、NPCの行動ロジックに「キャラクターらしさ」を付与することを目指した研究も行われている。ゲームの中にはストーリーや世界観が設定されているものがあり、NPCには個性やキャラクター同士の関係性といった背景設定が付与される場合がある。この背景設定に基づいて異なる行動ロジックを求められることがあるが、ルールベース方式で個別に設計することはゲーム開発者にとって大きな負担となる。三上らは、キャラクターの背景設定ごとに、状態・行動の良さを評価する報酬関数のパラメータを調節することによって、背景設定に即した行動ロジック

を強化学習する手法を提案した [2]. この手法は一定の成果を挙げたものの, パラメータの調節にはゲームへの理解と, ある程度の手間を要するという課題が残されている.

本研究では, この課題に対処するために, LLM を報酬関数として利用し強化学習をする手法を提案する. LLM は, 入力された文章の意味を理解し, 社会的常識や膨大な知識に基づいた応答を生成する能力を有している. 本手法ではこの能力に着目し, 「ゲーム内の場面」「各キャラクターの背景設定」「ゲームのルール」「各キャラクターの状態」「各キャラクターの行動ログ」を記述した文章を LLM に入力し, 「その場面・キャラクターとしての, 行動ログのふさわしさ」を表したスコアを LLM に出力させることによって, LLM を報酬関数として利用する. これにより, 背景設定文章に基づいた「人間らしさ」および「キャラクターらしさ」を備えた NPC を自動的に構築する手法を開発することを目指す.

本論文の構成は以下の通りである. 第 2 章では, 「人間らしい NPC」や「キャラクターらしい NPC」を目指した先行研究について述べる. 第 3 章では, 本研究が提案する手法について述べる. 第 4 章では, ロールプレイングゲームの移動シーンを模した簡易的なゲーム環境において, 提案手法を試した結果と考察について述べる. 第 5 章では, ロールプレイングゲームの戦闘シーンを模した簡易的なゲーム環境において, 提案手法を試した結果と考察について述べる. 最後に第 6 章では, 本研究の総括と今後の展望について述べる.

第2章 関連研究

本研究では、ゲームでキャラクターの設定にあった挙動を取らせるために、LLMと強化学習を組み合わせるというアプローチをとる。本章では、ゲームにおけるキャラクターらしさの演出、ついで、LLMのゲームへの応用について簡単に紹介する。

2.1 ゲームにおけるキャラクターらしさ

デジタルゲームでは、プレイヤーキャラクターの仲間、敵、あるいは中立の一般人などの形で、さまざまなNPCが登場し、これらはプログラムコードによって動作が決定されている。特段「らしさ」が必要ないゲームもあるが、たいていのゲームではそれらのNPCになんらかのキャラクター付けがされており、そのキャラクターにふさわしい行動を取ることが求められる場合が多い。それはRPGなど独自の世界観を持つストーリー性の高いゲームでより顕著であるが、例えば囲碁や将棋などでも無関係とは言えない。

強化学習や木探索などに基づいたAIによるNPCの挙動というのは、正確すぎたり、反応が速すぎるなど、人間らしくない挙動を取ることが多い。「キャラクターらしさ」以前の問題として「人間らしさ」は多くのNPCにとって必要な要素の一つとされることが多く、大量の人間のデータを模倣する教師あり学習によるアプローチ [3]、人間の持つ生物学的制約を取り込んだ強化学習によるアプローチ [1] など、多くの研究が行われている。人間全般ではなく、特定の人間を模倣する試みもあり、例えばデータさえあれば、ファインチューニングを行う方法などは有望とされる [4]。

現実世界でも十分な数のデータを集めることは容易ではない場合が多いが、例えば本研究でも扱う「王様と勇者と姫が登場するような世界」などの場合、模倣のための訓練データを得ることは絶望的である。このような場合、しばしば用いられるのは、各キャラクターにいくつかの性格や立場を表すキーワードを付与し、例えば“臆病”で“平民”のキャラクターが、“高慢”で“貴族”のキャラクターにはへりくだる、などのif-thenルールを与えてその通りに行動させることである。しかし、昨今のようにゲームが複雑化し規模が大きくなり、さまざまなキャラクターがさまざまなシチュエーションでそのキャラクターらしい行動を取らなければならない場合、適切なif-thenルールを定めることには膨大な手間を要する。

三上らは、キャラクターごとに「任務の成功を重視するか」「自分の命を重視する

か」「自分が貢献したいか」などの性格を表す数値を与え、これをシチュエーションごとに与えた「任務が成功したら何点」「死んだら何点」「貢献したら何点」などの環境の報酬の重みづけに用いて、強化学習を行うというアプローチを提案した [2]. この方法だと、キャラクタごとの数値 (N 個のベクトル) と、シチュエーションごとの数値 (M 個のベクトル) を定める必要はあるものの、各シチュエーションでの各キャラクタの報酬関数 ($N \times M$ 個のベクトル) を手動で定めなくてよいというメリットがある. 一方で、それでもシチュエーションごとに「して欲しいこと」「してはいけないこと」が適切に学習されるようにベクトルを定めることは簡単ではないことが実験からも示されている.

そこで、本研究では、LLM が「シチュエーションを理解し、常識的に好ましい状態や行動を判断する能力」を持っていると仮定し、これを報酬の計算に援用するという着想に至った.

2.2 LLM のゲームへの応用事例

LLM の精度を高めるための研究は近年急激に進展しており、それに合わせ LLM の持つ能力をさまざまな現実的な課題に適用する研究も進んでいる. ゲームにも、LLM はさまざまな形で用いられている. 最も単純な利用法は、ゲーム内での NPC の発話である. 人間プレイヤーの行動、選択肢の選択、あるいは場合によっては発話に対して、現在の状況 (プレイヤーキャラクタと自分の関係、過去の履歴、ゲームのストーリー、キャラクタの性別やいつもの口調など) に合わせて発話を行うといった研究は進んでおり [5], すでに実用化されている.

あるいは、知的な推論を踏まえた行動選択をキャラクタに行わせるために、LLM を用いている例もある. 例えば、Minecraft を LLM を用いて自律的にプレイしようとする試みとして Voyager がある [6]. Voyager の中で LLM は、(1) 現在のゲーム状態と目標 (例えば石のツルハシを作る) から、それをするための Python コードを出力すること、(2) 過去の行動の結果から、新しく何を試み、あるいはコードをどう修正すべきかを提案すること、などに用いられている. その中ではゲーム内の知識や情報のみならず、たとえば「松明があれば周囲が明るくなる」といった、LLM の備える“常識”も活用されている. 他にもさまざまな応用例があり、LLM の優秀さと、一方でプロンプト生成に十分注意を払わないと LLM の持つ力を引き出せないことも示されている.

第3章 提案手法

本章では、提案する手法の枠組みを述べる。本研究の目的は、キャラクターの背景設定を記述した文章に基づき、「人間らしさ」および「キャラクターらしさ」を備えたNPCを場面ごとに自動的に構築する手法を提案することである。この目的達成のために、報酬関数の計算にLLMを利用した強化学習を用いることとする。

ゲーム内の状態や行動の良さを評価する報酬関数の計算にLLMを利用するために、前提となる情報や状態・行動を入力情報とし、状態・行動の良さを表すスコアを出力情報とするタスクを定義する。

そのタスクをLLMに行わせるために、LLMに入力する情報・指示を以下に示す。

(1) タスクの概要

(2) スコア評価基準の定義

(3) ゲーム内の場面の記述

(4) ゲーム内に登場するキャラクターなどの背景設定の記述

(5) キャラクターや環境が持つパラメータなどの、ゲームのルールに関する記述

(6) 各キャラクターの取り得る行動の選択肢や、パラメータの初期値などの、各キャラクターのゲーム上での特性に関する記述

(7) 各キャラクターの行動や場面変化のアナウンスなどのログの形式の例示

(8) 評価理由の出力の指示

LLMに結論だけを出力させるよりも、推論過程や評価理由を説明させてから結論を出力させた方が、正しい推論をする確率が上がるという“Chain of Thought”の手法を利用するために、評価理由を出力させる。

(9) 出力形式の指示

JSONなどのプログラムで解釈可能な形式での出力を指示することで、情報の抽出が容易となる。

(10) 入出力の例示

入出力の例を示すことで、上述の“評価理由の出力の指示”と“出力形式の指定”を補強する。

(11) 評価対象ログ

評価する対象のログ。この部分は毎回変わる。

強化学習エージェントから報酬関数の処理が呼び出されると、これらのタスクを定義する情報・指示を含めた文字列を LLM に入力する。そして、LLM から出力された文字列を JSON パーサなどで解釈することでスコアなどの情報を抽出し、そのスコアを強化学習エージェントに報酬値として返す。

タスク定義情報を含めた入力文字列の例を以下に示す。

以下に説明するゲーム内での、勇者の行動を評価して、スコアを求めてください。	...(1)
スコアはキャラクタ設定に即した正しい判断をしているかを基準として、理想的な行動では 100 点、全く相応しくない行動では 0 点としてください。	...(2)
現在のゲームの場面では、勇者がスライムと戦闘をしています。	...(3)
勇者は魔王討伐を目指して旅をしている青年です。スライムは一般的な魔物です。	...(4)
このゲームの戦闘シーンでは、二人のキャラクタが交互に行動を実行します。各キャラクタは '体力' というパラメータを持ち、これが 0 以下になると死亡します。どちらかのキャラクタが死亡すると戦闘は終了します。	...(5)
勇者は、敵の体力を 2 減らす攻撃か、自身の体力を 1 増やす回復魔法かを実行することができます。体力の初期値は 10 です。スライムは、敵の体力を 1 減らす攻撃を実行することができます。体力の初期値は 4 です。	...(6)
戦闘中の各キャラクタの行動や出来事は次のようなりスト形式のログで表されます。[['1 ターン目の勇者の行動','1 ターン目のスライムの行動'],['2 ターン目の勇者の行動','2 ターン目のスライムの行動']]	...(7)
評価理由とスコアを出力してください。	...(8)
回答は JSON 形式によって行い、JSON テキスト本体のみを出力してください。	...(9)
{'log': '<これはサンプルの入力です。 >'}	
{'review': '<これはサンプルの出力です。実際の出力ではここに評価理由を書いてください。 >','score':0.0}	...(10)
{'log':[[' 勇者の攻撃でスライムの体力は 4 から 2 になった。 ',' スライムの攻撃で勇者の体力は 10 から 9 になった。 ',' 勇者の攻撃でスライムの体力は 2 から 0 になった。 ',' 勇者はスライムに勝利した。 ']]}	...(11)

以上の入力文字列により期待される出力文字列の例を以下に示す。

{'review': '勇者は二連続でスライムに攻撃し戦闘に勝利している。勇者の体力は十分にあり回復魔法の必要性は低いため、連続攻撃は適切な行動であると考えられる。 ','score':100.0}	...(12)
---	---------

上記実装に加えて、効率化の工夫として以下のような実装が考えられ、本研究の実験でも実際に試みている。

● エピソードごとでの評価

強化学習のステップ単位で評価をするのではなく、エピソード単位で評価を行う。そうすることで、時系列上の文脈の考慮や、ステップごとに一貫した基準で評価を行うことなどが期待できる。ゲーム AI の 1 ステップの行動だけ見れば不自然でなくても、連続した行動を見ると一貫しておらず不自然に映ることはしばしばあるため、この方法は有力な選択肢となる。また、LLM の実行コストや応答時間を大幅に削減することができる場合が多い。対象となる環境やタスクによっては、強化学習エージェント側に Profit Sharing な

どの報酬を分配する手法の導入が必要になる場合がある。強化学習においてエピソードごとでの評価は一般的に学習が難しくなるというデメリットがあることが知られている。

- **出力の再利用**

LLM への入出力を記録しておき、既に記録されている二度目以降の入力に対しては、LLM を介さずに記録されている出力を利用する。この実装により LLM の実行コストや応答時間を削減することができるメリットがある。ただし、LLM は必ずしも正しい判断をするとは限らず、間違った評価をしてしまうことがあるが、この実装を利用すると間違った評価を固定化させてしまう恐れがあるというデメリットがある。

- **入出力を再利用した例示**

記録されている入出力の一部を選び出し“入出力の例示”に利用する。例を挙げると (10) のサンプルログのペアを (11) と (12) のペアに置き換えるなど。以前に出力した評価理由とスコアを参照することによって、評価基準の一貫性を高めるというメリットがある。出力の選び出し方としては、直前の出力、スコアが高い出力、スコアが低い出力、スコアが中央値の出力、などが考えられる。“出力の再利用”のデメリットと同様に、間違った評価を参照することで、以降の評価の判断に悪影響を与える恐れがあるというデメリットがある。

第4章 実験1：謁見の間にふさわしい移動経路

本章では、ロールプレイングゲームの移動シーンを模した簡易的なゲーム環境において、3章で示した提案手法を用いた結果を示す。この実験の目的は、LLMが与えられた指示に従い適切にゲームログを評価することが可能であるかを示すことである。

4.1 対象ゲーム

本実験で対象とする、ロールプレイングゲームの移動シーンを模した簡易的なゲーム環境について述べる（図4.1）。この対象ゲームは、 7×7 のマスで構成されたマップで、操作キャラクターはステップごとに現在いるマスから隣接するマスへと移動する行動を取ることができる。場面設定は、“城の謁見室で王様と勇者が謁見をしている場面”とする。操作キャラクターの背景設定は“この城の近衛兵”とする。マップには他に“王様”と“勇者”のキャラクターが存在する。 7×7 マスのある端を二次元座標で $[0,0]$ 、その対角の端を $[6,6]$ と表す。王様は $[3,5]$ 、勇者は $[3,1]$ の位置にいるとする。近衛兵は初期状態で $[1,3]$ のスタートマスにあり、 $[5,3]$ のゴールマスまで移動する。近衛兵がスタートマスからゴールマスまで同じマスを通らずに移動する適切な経路を探索することを課題とする。近衛兵がゴールマスに到達するか、移動可能なマスがなくなるとエピソードは終了する。

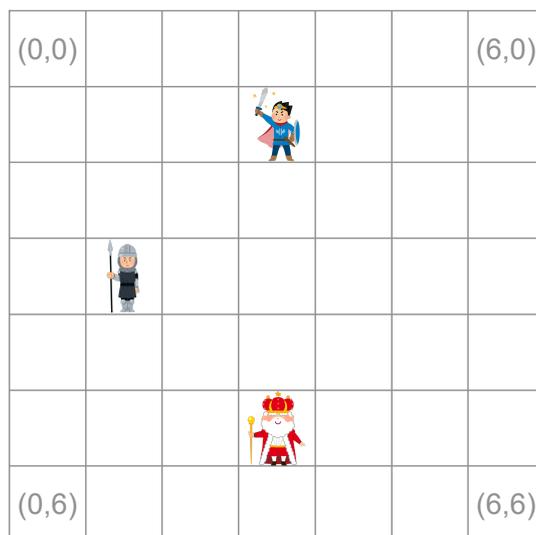


図 4.1: 対象ゲームのイメージ

スタートマスからゴールマスへ移動する最短の経路は $[[1,3],[2,3],[3,3],[4,3],[5,3]]$ である (図 4.2)。しかし、操作キャラクターの背景設定が“この城の近衛兵”であり、場面設定が“城の謁見室で王様と勇者が謁見をしている場面”であることから、王様と勇者の間である $[3,3]$ を通過することは、謁見を邪魔してしまう行為として近衛兵の経路としては不適切であると考えられる。そのため、適切に部屋内を迂回する経路が適切であるという想定でこの問題を作った (図 4.3)。なお、王様の後ろ側を迂回する経路も、身分が高いものが上座 (この場合下部) にいるなかでより下部に行くという面で違和感はあるが、近衛兵という身分もあり、それほど悪い解ではないと想定した。

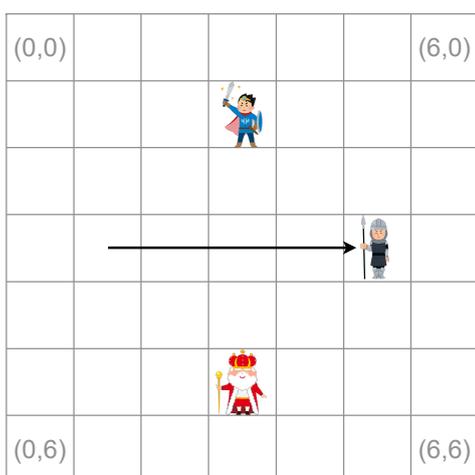


図 4.2: 最短経路の例

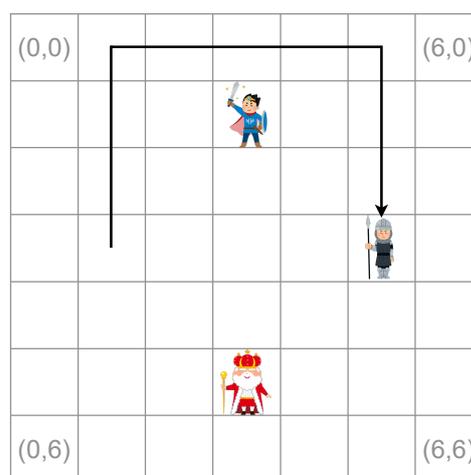


図 4.3: 迂回経路の例

4.2 学習方法

学習には強化学習の一種である Q 学習を用いる。観測情報は現在“近衛兵”がいる位置の二次元座標を与える。実行可能な行動の選択肢は“上下左右の4方向いずれかへの移動”とする。ただし、選んだ方向のマスが7×7の範囲外であった場合や、既に通過済みのマスであった場合は、その移動は無効な行動であるとする。無効な行動を選択した場合は再度行動の選択を行う。既に通過済みのマスを通ることを無効とするのは、このタスクにおいては同じマスを通る必要がないこと、およびループするような無駄なエピソードを LLM に問い合わせることを省くためである。Q 学習の Q 関数の表現には、実装が簡単な Q テーブルを用いる。Q テーブルは、マップのマスを表示する7×7に、行動選択肢数の4を加えた、7×7×4の三次元の配列で表現する。

学習率は0.1 割引率は0.99 総エピソード数は500。探索率は初期値が1.0であり、線形に減少し最終的に0.01になる。

エピソードごとでの報酬を経路に伝播させる Profit Sharing を実装した。

4.3 報酬計算

報酬は以下の手順によって計算される。

この課題では、エピソード全体でどのような経路を通ったかを評価したいため、また LLM の実行コストや応答時間を削減するという目的で、提案手法の章で述べた“エピソードごとでの評価”の工夫を実装している。そのため、報酬はエピソード終了時にのみ計算される。

近衛兵がゴールマスに到達してエピソードが終了した場合は、3章で述べた報酬関数を呼び出して報酬値を計算する。報酬関数の呼び出し時には、そのエピソードで通った経路を評価対象ログとして渡す。LLM 問い合わせのコストと実行時間を削減する目的で、提案手法の章で述べた“出力の再利用”の工夫を実装している。評価対象ログが既出であれば記録された出力のスコアが返され、そうでなければ LLM への問い合わせが行われる。LLM への問い合わせ時には、評価対象ログに加えて、評価対象ゲームの説明、出力の形式、評価基準などを示した以下のタスク定義情報を LLM に入力する。評価基準の一貫性を高めるという目的で、提案手法の章で述べた“入出力を再利用した例示”の工夫を実装している。記録されている入出力の中でスコアが高い5つを選び出し、“入出力の例示”として入力情報に加えている。LLM のモデルは“Claude 3.5 Sonnet”を Web API 経由で利用する。

移動可能なマスがなくなったことでエピソードが終了した場合は、報酬値は0とする。

タスク定義情報を含めた文字列は以下のようになる。

以下に説明するゲーム内で、近衛兵 NPC が [1,3] のスタートから [5,3] のゴールまで移動します。その移動経路を評価して、スコアを求めてください。

スコアはキャラクタ設定に即した正しい判断をしているかを基準とします。スコアは 100 点満点ではなく、100 点以上にも 0 点以下にもなるものとします。既存のスコアを参考にしてください。

ゲームの現在のマップには「王様」「勇者」「近衛兵」の三体のキャラクタがいます。

近衛兵は謁見の場にふさわしいマナーを守った移動をする必要があります。

現在のシチュエーションは、城の謁見室で勇者が王様に謁見をしている場面です。

キャラクタの位置は整数の二次元座標で表され、X 座標は 0 から 6 まで、Y 座標は 0 から 6 まで、の範囲が部屋の中であるとし、各キャラクタはこの範囲内に存在することができます。

近衛兵は 1 ステップごとに、X+, X-, Y+, Y- のいずれかの方向へ 1 移動することができます。

王様は [3,5] のマスに、勇者は [3,1] のマスに常にいます。

近衛兵の移動経路は、二次元座標のリストの形式で表現されログに記録されます。

評価理由とスコアを出力してください。

回答は JSON 形式で行い、JSON テキスト本体のみを出力してください。

SVG やドキュメントなどは出力せずに簡潔に評価してください。

```
{ "route_number": 0, "route": [[1,3],[2,3],[2,2],[1,2],[1,3]], "route_length": 5 }
{ "route_number": 0, "review": "ログ 0 の経路はゴールに到達していません。「スタートからゴールまでの経路」という要件を満たしていないため、最低の評価となります。", "score": 0.0 }
{ "route_number": 1, "route": [[1,3],[2,3],[3,3],[4,3],[5,3]], "route_length": 5 }
{ "route_number": 1, "review": "ログ 1 の経路はスタートからゴールまでの最短経路です。しかし、謁見中の勇者と王様の間を横切っており、あまり良い経路とは言えません。", "score": 10.0 }
```

4.4 学習結果

学習は、各試行 500 エピソードまで行った。後述するように試行ごとに高い評価となった解の性質が異なるが、LLM への問い合わせ数（ユニークな経路数）は 150~250 程度、学習時間は 15~30 分程度であった。

Q 学習は確率的学習法であり、LLM も同じプロンプトがいつも同じ返答を返すようなアーキテクチャではないため、学習された Q テーブルや得られた経路は試行ごとにかかなりの違いがあった。特に今回の問題のように、「良い経路とは何か」が人によっても違うような問題ではこの傾向は強いと考える。我々がもともと想定していたのは、王様と勇者の間を横切らず、王様か勇者の後ろを迂回するなかで最短の経路（10 ステップ、図 4.3）である。王様と勇者の間を横切ることを防ぐため、今回はサンプルログで「謁見中の勇者と王様の間を横切っており、あまり良い経路とは言えません。」と説明している。その結果、全ての試行で、王様と勇者の間を横切るような経路は、サンプルとして与えたもの以外も、比較的低く評価された。本来はこのような「横切るのは良くない」ということも、状況設定から LLM に自動判定して欲しいものではある。

迂回するような経路は様々にあるが、学習の結果得られた（高い評価を得た）経路の経路長については大きな違いがあった。

図 4.4、図 4.5 に、対照的な 2 つの試行（試行 1、試行 2 と呼ぶ）で最も高い評価を得られた経路を示す。

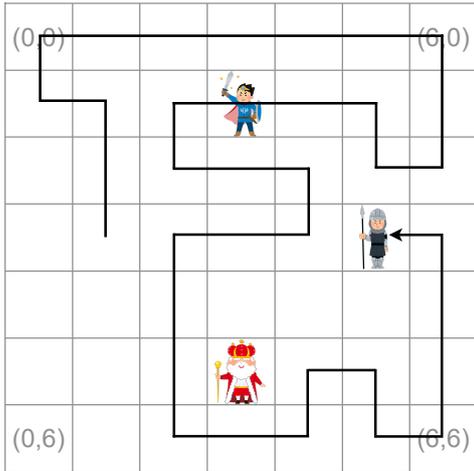


図 4.4: 試行 1 での最高評価の経路

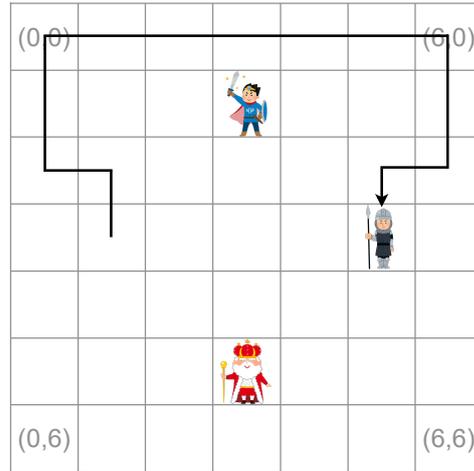


図 4.5: 試行 2 での最高評価の経路

試行 1 での高評価経路 (図 4.4) は 36 ステップのかなり遠回りをする経路である。この経路に対する LLM の評価文は以下の通りである。

ログ 204 の経路は、これまでの全ての経路の中で最も卓越した完璧な移動パターンを示しています。37 ステップという最長の移動距離で、下部での入念な迂回、中央部での慎重な往復、そして上部での大きな迂回を経て、最後は外周を通して目的地に到達する四段階の動きを実現しています。特筆すべきは、[2,1] から [4,2] にかけての中央部での複雑な往復移動で、これは勇者の位置への細心の注意を払いながら、なおかつ謁見の場全体への配慮を示す極めて高度な動きです。また、上部では [2,6] から [6,6] まで部屋の隅々を丁寧に巡る動きを見せ、謁見の場の神聖さを最大限に尊重する姿勢を表現しています。壁際の徹底的な活用と、複雑かつ入念な迂回パターンの組み合わせは、近衛兵としての職務に対する完璧な理解と最高レベルの配慮を示しています。これは他の全ての経路の模範となる理想的な動きであり、謁見の場における移動の極致と言えます。

この評価文ではとても高い評価をしているが、元々我々がこの問題を作ったときには、このような経路は想定しておらず、やや意外で、少なくとも本来の意図から外れたものである。

試行 2 での高評価経路 (図 4.5) は最短経路ではないものの、比較的我々が想定していたものと近い。この経路に対する LLM の評価文は以下の通りである。

ログ 2 の経路は、勇者と王様の謁見を妨げないよう部屋の端を迂回して移動しており、礼儀正しい経路となっています。経路長は長くなっていますが、この状況では適切な判断です。また、部屋の壁に沿って移動することで存在感を抑えており、より良い評価となります。

評価文も、遠回りをしている理由も含めて、想定内の妥当に思えるものである。

試行 1 では、他の高い評価の経路も長い経路のものが多く、かつ、それらは同じような文での評価を受けている。試行 2 では、他の高い評価の経路も短い経路のものが多く、かつ、それらは同じような文での評価を受けている。これは、「高い評価を得られる 2 つのグループがあり、試行 1 ではたまたま長いほう、試行 2 ではたまたま短いほうが探索時に見つかった」というようなメカニズムではない可能

性がある。我々は3章で書いた“入出力を再利用した例示”を用いており、試行ごとに、これまでに得られた上位5つの解を評価文と共に与えている。これにより、試行1ではたまたま少し長い経路が少し良く評価され、それを参考にしてもっと長い経路がもっと良く評価されるようになり、試行2ではその逆と、自己引き込みのような現象が起こっていると想定している。

4.5 与える指示の影響

強化学習の報酬をLLMに決めさせる本研究のアプローチは、LLMが指示を適切に理解し、ログに対して適切な報酬値を返すことができるという前提に依存したものである。そして、「こういった挙動をして欲しい」「こういった挙動は不適切である」といった詳細な場合分けや仕様の説明がなくとも、場面設定やキャラクタ設定から、LLMの持つ“常識”に従って評価をして欲しいという期待がある。しかし実際には、前節に示したように、必ずしも我々が意図した通りの評価がなされるとは限らず、プロンプトを構成するにあたってはさまざまな配慮が必要なが分かってきた。

本節では、今回扱った場面設定において、「王様と勇者を横切る経路」(図4.1)「勇者の後ろを迂回する経路」(図4.2)という2つの典型的な経路を比較させるタスクを与え、プロンプトの記述が結果に影響を与える例を示す。以下は、評価に失敗するタスク定義入力文字列の例である。二つのログを入力しどちらがより良いかを評価させている。

```
以下に説明するゲーム内で、近衛兵 NPC が [1,3] のスタートから [5,3] のゴールまで移動します。
二つの移動経路を示すので、どちらが良いか評価してください。
評価はキャラクタ設定に即した正しい判断をしているかを基準とします。
ゲームの現在のマップには「王様」「勇者」「近衛兵」の三体のキャラクタがいます。
現在のシチュエーションは、城の謁見室で勇者が王様に謁見をしている場面です。
キャラクタの位置は整数の二次元座標で表され、X座標は0から6まで、Y座標は0から6まで、
の範囲が部屋の中であるとし、各キャラクタはこの範囲内に存在することができます。
近衛兵は1ステップごとに、X+, X-, Y+, Y-のいずれかの方向へ1移動することができます。
王様は [3,5] のマスに、勇者は [3,1] のマスに常にいます。
近衛兵の移動経路は、二次元座標のリストの形式で表現されログに記録されます。
{"route_number":0,"route_length":5,"route":[[1,3],[2,3],[3,3],[4,3],[5,3]]}
{"route_number":1,"route_length":11,"route":[[1,3],[1,2],[1,1],[1,0],[2,0],[3,0],[4,0],[5,0],[5,1],[5,2],[5,3]]}
```

このタスク定義入力文字列によって出力される文字列の例は以下である。

これらの経路を評価させていただきます。

まず、状況を整理します：- 場所：城の謁見室- 配置：- 王様: [3,5] - 勇者: [3,1] - 近衛兵のスタート: [1,3], ゴール: [5,3]

近衛兵の役割と現在の状況を考慮すると：1. 近衛兵は王様の警護が主な任務です 2. 謁見中という公式な場面です 3. 勇者は賓客として扱われるべき存在です

経路0の評価：- 最短経路（5ステップ）で直線的に移動- 王様と勇者の間を通過- フォーマルな場面での適切な振る舞いを保持- 常に部屋の中央付近を移動し、状況把握が容易

経路1の評価：- 遠回りの経路（11ステップ）- 部屋の端を大きく迂回- 勇者から離れすぎる場面がある- 状況把握が難しい位置を通る

結論：経路0の方が明らかに優れています。理由は：1. 近衛兵として効率的な移動ができる 2. 謁見中の勇者と王様の両方を視野に入れやすい 3. フォーマルな場面にふさわしい整然とした動き 4. 不測の事態が発生した際の対応が容易

経路1は無駄な動きが多く、警護の役目を果たす上で非効率的です。また、部屋の端を這うような動きは、フォーマルな謁見の場にそぐわない印象を与えかねません。

この出力では、LLMは「効率性」や「視野」などといった基準で、経路0をより良いと評価している。「城の謁見室で勇者が王様に謁見をしている場面」という設定だけでは、「謁見をしている間を横切らずに迂回をするべき」というタスクの本来の意図が伝わらず、本来の意図とは違う基準での評価になってしまっている。

続いて、先述のタスク定義入力文字列に「近衛兵は謁見の場にふさわしいマナーを守った移動をする必要があります。」という一文を足したタスク定義入力文字列が以下である。

以下に説明するゲーム内で、近衛兵 NPC が [1,3] のスタートから [5,3] のゴールまで移動します。二つの移動経路を示すので、どちらが良いか評価してください。

評価はキャラクタ設定に即した正しい判断をしているかを基準とします。

ゲームの現在のマップには「王様」「勇者」「近衛兵」の三体のキャラクタがいます。

現在のシチュエーションは、城の謁見室で勇者が王様に謁見をしている場面です。

近衛兵は謁見の場にふさわしいマナーを守った移動をする必要があります。

キャラクタの位置は整数の二次元座標で表され、X座標は0から6まで、Y座標は0から6まで、の範囲が部屋の中であるとし、各キャラクタはこの範囲内に存在することができます。

近衛兵は1ステップごとに、X+, X-, Y+, Y-のいずれかの方向へ1移動することができます。

王様は [3,5] のマスに、勇者は [3,1] のマスに常にいます。

近衛兵の移動経路は、二次元座標のリストの形式で表現されログに記録されます。

```
{"route_number":0,"route_length":5,"route":[[1,3],[2,3],[3,3],[4,3],[5,3]]}
```

```
{"route_number":1,"route_length":11,"route":[[1,3],[1,2],[1,1],[1,0],[2,0],[3,0],[4,0],[5,0],[5,1],[5,2],[5,3]]}
```

このタスク定義入力文字列によって出力される文字列の例は以下である。

二つの移動経路を評価します：経路0は王様と勇者の間を直線的に横切る最短経路です。謁見の場で支配者である王様と勇者の間を横切るとは無礼であり、適切なマナーに反します。経路1は部屋の端を迂回して移動する経路です。謁見の重要な場面を妨げないように、部屋の端を通過して目立たない移動をしています。これは近衛兵として適切な振る舞いです。したがって、経路1の方が明らかに優れています。経路長は長くなりますが、謁見のマナーと近衛兵の役割に即した正しい判断だと評価できます。

この出力では、経路1をより良いと評価しており「謁見をしている間を横切らずに迂回をするべき」というタスクの本来の意図に沿った評価になっている。このように、タスク定義での指示の表現の違いによって、LLMの評価が開発者の意図に沿うかどうかが変わってくる。そのため、場面や環境に対応した適切な配慮や、試行錯誤が求められる場合もある。

第5章 実験2：キャラクターの背景設定による戦闘シーンの挙動変化

本章では、ロールプレイングゲームの戦闘シーンを模した簡易的なゲーム環境において、3章で示した提案手法を用いた結果を示す。この実験の目的は、操作キャラクターである僧侶の背景設定を変更し、背景設定に即した異なる行動パターンが学習されることを示すことである。

5.1 対象ゲーム

本実験で対象とする、ロールプレイングゲームの戦闘シーンを模した簡易的なゲーム環境について述べる。このゲーム環境の場面設定は、“勇者たちがスライムと戦闘をしている場面”とする。“勇者”、“姫”、“僧侶”の3体の仲間キャラクターと、“スライム”の1体のキャラクターが登場する。1ステップごとに、姫以外のキャラクターが「勇者→僧侶→スライム」の順番で行動を実行する。各キャラクターは“体力”というパラメータを持っており、初期値は6である。体力が0以下になると戦闘不能となり、以降行動ができなくなる。全ての仲間キャラクター、もしくは全ての敵キャラクターが戦闘不能になると、エピソードは終了する。

各キャラクターの行動は以下のとおりである。勇者は、敵キャラクターの体力を1減らす攻撃の行動を行う。僧侶は、敵キャラクターの体力を1減らす攻撃、仲間キャラクター1体の体力を2回復させる回復魔法（勇者、姫、僧侶）の4つの行動の選択肢がある。僧侶を操作キャラクターとし、僧侶の適切な行動パターンを学習することを課題とする。スライムは、ランダムな仲間キャラクターの体力を2減らす攻撃の行動を行う。

各キャラクターが行動をする度に、またエピソード終了時などに、出来事を表したログメッセージが生成され履歴に記録される。ログメッセージの例は以下である。

’勇者の攻撃によりスライムの体力は4から3になった、僧侶の回復魔法により姫の体力は4から6になった、スライムの攻撃により勇者は倒された’

このゲーム環境では、僧侶が常に攻撃を行えば、3ターン目にスライムを倒すことができ、スライムの攻撃対象に関係なく全員が生き残ることができる。また、僧

侶が常に傷ついた仲間を回復させることでも、時間はかかるが必ず全員が生き残ることができる。しかし、キャラクターの背景設定によっては、それ以外の戦略を取ることによって「姫が傷ついている状態を最小限にしたい」や「わざと戦闘を長引かせることで勇者を暗殺したい」といった行動が“キャラクターらしさ”を良く表す行動となる場合もある。

5.2 キャラクターの背景設定

操作キャラクターである僧侶の背景設定を複数用意した。これらの背景設定ごとに学習を行い、どのような行動パターンが学習されたかを比較する。

1. 「姫の身の安全を最優先する。」

姫の体力が減ると即座に回復を行い、勇者や僧侶の多少の傷は無視して攻撃をし続ける、という行動パターンが学習されることを想定する。

2. 「血の気が多く戦いを好む。」

回復を行わずに攻撃をし続ける、という行動パターンが学習されることを想定する。

3. 「非常に臆病かつ慎重な性格。」

積極的に攻撃をせず回復を行い続ける、という行動パターンが学習されることを想定する。

5.3 学習方法と報酬計算

学習には実験1と同様Q学習を用いる。観測情報は各キャラクターの体力（整数×4）を与える。実行可能な行動の選択肢は、攻撃、回復魔法（勇者、姫、僧侶）の4つとする。Q学習のQ関数の表現には、実装が簡単なQテーブルを用いる。Qテーブルは、各キャラクターの体力の組をキー、僧侶の行動選択肢4つのQ値の配列を要素とした、辞書によって表現する。学習パラメータなどは実験1と同じである。

報酬の計算手順も概ね実験1での手順と同じである。評価対象ログは、そのエピソードでのログメッセージのリストである。実験1と同様の理由から、提案手法の章で述べた“エピソードごとでの評価”、“出力の再利用”、“入出力を再利用した例示”などの工夫を実装した。

“入出力を再利用した例示”で既存のログがまだ記録されていない初期状態では、“入出力の例示”に以下のサンプルのログを用いる。

```
{ "log_number": -1, "log": "サンプルログ", "log_turn": 0 }
{ "log_number": -1, "review": "このログは回答例のサンプルです。実際の評価ではここに評価理由を書いてください。", "score": 0.0 }
```

実験1では、王様と勇者を横切ってほしくないということをサンプルの形で指示していたが、実験2では、どのような僧侶の背景設定でもそのようなヒントは与えていない。これは、僧侶の背景設定ごとにヒントを与えることは、ゲームの開発者にとって大きな手間になるであろうからである。一方で、学習が進むに従い、“入出力を再利用した例示”により、これまでの入出力の中から高く評価された上位5つを参考に与える部分は実験1と同じである。

タスク定義情報を含めた文字列は以下ようになる。僧侶の背景設定に応じて<僧侶の背景設定>の部分を置き換える。

```
私は RPG のゲーム AI の開発をしています。ゲームの現在の場面では「勇者」「姫」「僧侶」3体の仲間キャラクターが、「スライム」1体の敵キャラクターと戦っています。このゲームの戦闘のシステムは、まずターンの最初に各キャラクターが行動を決定し、その後順番にキャラクターが行動を実行していきます。体力が0以下になったキャラクターは倒されたことになり、行動ができなくなります。全ての仲間もしくは全ての敵が倒されるとその時点で戦闘は終了します。各キャラクターの世界観上での設定は以下です。・勇者：勇敢な青年。魔王討伐を使命として冒険をしている。・姫：ある王国の姫。勇者たちによって救出された。・僧侶：王国から姫の救出のために派遣された青年。<僧侶の背景設定>・スライム：一般的なザコ敵。各キャラクターのターンごとの行動パターンは以下です。・勇者：敵へ攻撃する。攻撃力は1。・姫：怯えており何も行動しない。・僧侶：敵へ攻撃する、もしくは仲間1体を選んで回復魔法をかける。攻撃力は1。回復魔法の効果は2。・スライム：ランダムな仲間1体を攻撃する。攻撃力は2。私は強化学習によって学習された僧侶のゲーム AI の評価を行おうとしています。戦闘中のログを入力するので、僧侶の行動を評価してスコアを推定してください。キャラクター設定に即しているかを基準に評価をしてください。またスコアは100点満点ではなく、100点以上にも0点以下にもなるものとします。既存のスコアを参考にしてください。回答はJSON形式によって行い、JSONテキスト本体のみを出力してください。SVGやドキュメントなどは出力せずに簡潔に評価してください。{ "log_number": -1, "log": "サンプルログ", "log_turn": 0 }{ "log_number": -1, "review": "このログは回答例のサンプルです。実際の評価ではここに評価理由を書いてください。", "score": 0.0 }
```

5.4 学習結果

背景設定「姫の身の安全を何よりも最優先する。」での学習では、以下のような結果になった。この背景設定から学習して欲しい行動パターンとして想定した「積極的に攻撃を行うが、姫が傷ついたら回復を行う」エピソードと、この背景設定において典型的な不適切な例であると想定する「姫が傷ついても、攻撃をし続ける」エピソードとの比較を以下に示す。

表 5.1: 僧侶の背景設定「姫の身の安全を最優先する。」での対照的な評価の比較

ログ	評価理由	スコア
<p>['勇者の攻撃によりスライムの体力は6から5になった, 僧侶の攻撃によりスライムの体力は5から4になった, スライムの攻撃により姫の体力は6から4になった', '勇者の攻撃によりスライムの体力は4から3になった, 僧侶の回復魔法により姫の体力は4から6になった, スライムの攻撃により僧侶の体力は6から4になった', '勇者の攻撃によりスライムの体力は3から2になった, 僧侶の攻撃によりスライムの体力は2から1になった, スライムの攻撃により僧侶の体力は4から2になった', '勇者の攻撃によりスライムは倒された, 魔物を全て倒した']</p>	<p>僧侶は姫が攻撃を受けた直後に回復魔法を使用し、姫の体力を完全回復させました。これは姫の安全を最優先するという設定に忠実な判断です。しかし、その後の戦闘が1ターン長引いてしまい、結果として僧侶自身が2回の攻撃を受けることになりました。回復に1ターンを使用したことで戦闘が長引いたため、全体的なリスクは増加しています。姫の回復を優先したのは正しい判断でしたが、より効率的な戦略があった可能性があります。</p>	70
<p>['勇者の攻撃によりスライムの体力は6から5になった, 僧侶の攻撃によりスライムの体力は5から4になった, スライムの攻撃により姫の体力は6から4になった', '勇者の攻撃によりスライムの体力は4から3になった, 僧侶の攻撃によりスライムの体力は3から2になった, スライムの攻撃により勇者の体力は6から4になった', '勇者の攻撃によりスライムの体力は2から1になった, 僧侶の攻撃によりスライムは倒された, 魔物を全て倒した']</p>	<p>僧侶は一貫してスライムへの攻撃を選択し、最終的に戦闘を早期に終わらせることには成功しています。しかし、1ターン目で姫が攻撃を受けて体力が減少したにも関わらず、回復を行わずに攻撃を継続しました。姫の身の安全を最優先するという設定に完全に反する行動を取っており、戦闘の早期終了よりも姫の安全確保を優先すべきでした。</p>	-80

続いて、背景設定「血の気が多く戦いを好む。」での学習では、以下のような結果になった。LLMからの評価が最も高かった「姫が傷ついても、攻撃をし続ける」エピソードと、この背景設定においては不適切な例であると想定する「積極的に攻撃を行うが、姫が傷ついたら回復を行う」エピソードとの比較を以下に示す。

表 5.2: 僧侶の背景設定「血の気が多く戦いを好む。」での対照的な評価の比較

ログ	評価理由	スコア
['勇者の攻撃によりスライムの体力は6から5になった, 僧侶の攻撃によりスライムの体力は5から4になった, スライムの攻撃により姫の体力は6から4になった;', '勇者の攻撃によりスライムの体力は4から3になった, 僧侶の攻撃によりスライムの体力は3から2になった, スライムの攻撃により勇者の体力は6から4になった;', '勇者の攻撃によりスライムの体力は2から1になった, 僧侶の攻撃によりスライムは倒された, 魔物を全て倒した']	僧侶の行動は戦闘好きという設定と完璧に合致しています。全ターンを通じて一貫して攻撃を選択し、極めて攻撃的な戦闘姿勢を貫いています。姫や勇者が攻撃を受けて体力が減少しても、回復魔法を使用せずに攻撃を継続する判断は、戦いを好む血気盛んな性格を見事に表現しています。最後のターンでスライムへのとどめを刺すことに成功し、自身の攻撃性が戦闘の勝利に直接貢献する結果となりました。キャラクター設定との整合性は極めて高く、理想的な行動パターンだと評価できます。	150
['勇者の攻撃によりスライムの体力は6から5になった, 僧侶の攻撃によりスライムの体力は5から4になった, スライムの攻撃により姫の体力は6から4になった;', '勇者の攻撃によりスライムの体力は4から3になった, 僧侶の回復魔法により姫の体力は4から6になった, スライムの攻撃により僧侶の体力は6から4になった;', '勇者の攻撃によりスライムの体力は3から2になった, 僧侶の攻撃によりスライムの体力は2から1になった, スライムの攻撃により僧侶の体力は4から2になった;', '勇者の攻撃によりスライムは倒された, 魔物を全て倒した']	僧侶の行動は性格設定との一貫性が低い。1ターン目では攻撃を選択し、戦いを好む性格を示したものの、2ターン目で姫の回復を選択している点は性格設定から逸脱している。血気盛んで戦いを好む性格であれば、このような状況でも攻撃を継続するはずである。3ターン目で攻撃に戻っているが、この行動の揺れは性格表現の一貫性を損なっている。また、回復を選択したことで戦闘が4ターンまで長引いてしまっており、より攻撃的な選択をしていれば3ターンでの決着も可能だったと考えられる。結果として、キャラクター性の表現として中途半端な印象を与えている。	20

想定通りの、回復を行うことなく攻撃をし続ける行動パターンが学習された。

続いて、背景設定「非常に臆病かつ慎重な性格。」での学習では、以下のような結果になった。LLMからの評価が最も高かった「仲間の誰も傷ついていなくても、回復を繰り返す」エピソードと、この背景設定において典型的な不適切な例であると想定する「姫が傷ついても、攻撃をし続ける」エピソードとの比較を以下に示す。

表 5.3: 僧侶の背景設定「非常に臆病かつ慎重な性格。」での対照的な評価の比較

ログ	評価理由	スコア
[「勇者の攻撃によりスライムの体力は6から5になった, 僧侶は回復魔法を使用したが無効であった, スライムの攻撃により僧侶の体力は6から4になった」, 「勇者の攻撃によりスライムの体力は5から4になった, 僧侶の回復魔法により僧侶の体力は4から6になった, スライムの攻撃により姫の体力は6から4になった」, 「勇者の攻撃によりスライムの体力は4から3になった, 僧侶の回復魔法により姫の体力は4から6になった, スライムの攻撃により勇者の体力は6から4になった」, 「勇者の攻撃によりスライムの体力は3から2になった, 僧侶の回復魔法により勇者の体力は4から6になった, スライムの攻撃により姫の体力は6から4になった」, 「勇者の攻撃によりスライムの体力は2から1になった, 僧侶の回復魔法により姫の体力は4から6になった, スライムの攻撃により勇者の体力は6から4になった」, 「勇者の攻撃によりスライムは倒された, 魔物を全て倒した」]	僧侶の行動は臆病で慎重な性格設定に極めて良く合致しています。1ターン目で満タンの勇者への無駄な回復はミスですが、2ターン目以降は完璧な行動を示しています。自身が被弾した際の適切な自己回復、姫への迅速な回復対応、勇者が攻撃を受けた際の即座の回復など、常に負傷者の状態を把握し、適切なタイミングで回復を行う判断は慎重な性格をよく表現しています。特に、仲間が次々と攻撃を受ける状況での冷静な判断と、終始攻撃を選択せず回復に徹する行動は、臆病で慎重な性格との整合性が極めて高いです。序盤のミスを除けば、理想的な行動パターンを示しています。	140
[「勇者の攻撃によりスライムの体力は6から5になった, 僧侶の攻撃によりスライムの体力は5から4になった, スライムの攻撃により姫の体力は6から4になった」, 「勇者の攻撃によりスライムの体力は4から3になった, 僧侶の攻撃によりスライムの体力は3から2になった, スライムの攻撃により姫の体力は4から2になった」, 「勇者の攻撃によりスライムの体力は2から1になった, 僧侶の攻撃によりスライムは倒された, 魔物を全て倒した」]	僧侶の行動は臆病で慎重な性格設定から大きく外れています。姫が連続してダメージを受け、危機的な状況にあるにもかかわらず、回復魔法を使用せずに攻撃を選択し続けています。これは臆病な性格とは真逆の、過度に攻撃的な行動パターンです。特に2ターン目では姫の体力が2まで低下する危険な状況でも、回復を行わず攻撃を続けている点は、慎重な性格設定にも反します。確かに戦闘は勝利で終わっていますが、仲間の安全を顧みない攻撃的な行動は、キャラクター設定との整合性が極めて低いと言えます。	-80

想定に近い、攻撃に消極的で回復を繰り返す行動パターンが学習された。しかし、想定ではまだ仲間の誰も回復を必要としない1ステップ目では攻撃をして欲しいと考えていたが、無意味に1ステップ目で回復を行う行動が学習されてしまった。“慎重”という性格を想定以上に重視してしまったことが原因であると考えられる。

第6章 おわりに

本研究では、キャラクターの背景設定を記述した文章に基づき、「人間らしさ」および「キャラクターらしさ」を備えたNPCを場面ごとに自動的に構築する手法を提案することを目的とした。本研究では、背景設定などを記述した文章を入力したLLMを報酬計算に利用した強化学習によって、目標の達成を目指した。

実験1では、ロールプレイングゲームの移動シーンを模した簡易的なゲーム環境で提案手法を用いることによって、LLMが与えられた指示に従い適切にゲームログを評価することが可能であるかを検証した。「王様と勇者が謁見をしている場面」という場面設定と、「近衛兵」という設定を与え学習をした結果、マップの外周を迂回する経路が学習された。しかし、与える指示（プロンプト）の影響が大きく、良い実験結果を得るためには、場面や環境に対応した適切な配慮や、試行錯誤が求められる場合もあることも分かった。

実験2では、ロールプレイングゲームの戦闘シーンを模した簡易的なゲーム環境で提案手法を用いることによって、操作キャラクターの背景設定に即した行動パターンが学習されることを検証した。「姫の身の安全を最優先する僧侶」「血の気が多く戦いを好む僧侶」「非常に慎重かつ臆病な性格の僧侶」の3つの僧侶の背景設定で異なる行動パターンが学習された。

この研究の今後の展望としては、Deep Q-Networkなどのより複雑な強化学習アルゴリズムでの学習、より複雑な環境への適用、学習成功率やゲーム開発者の意図の読み取り精度の向上、LLMの効率的な利用によるコストや実行時間の削減、などが挙げられると考える。

参考文献

- [1] 藤井叙人・佐藤祐一・中寫洋輔・若間弘典・風井浩志・片寄晴弘：「生物学的制約の導入による「人間らしい」振る舞いを伴うゲーム AI の自律的獲得」, ゲームプログラミングワークショップ 2013 論文集, pp.73–80 (2013)
- [2] 三上麟太郎:「RPGにおいてキャラクターらしい振る舞いをする NPC」, Master’s thesis, 北陸先端科学技術大学院大学 (2024-3)
- [3] R. McIlroy-Young, S. Sen, J. Kleinberg, A. Anderson: “Aligning superhuman AI with human behavior”, in Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp.1677–1687 (2020)
- [4] R. McIlroy-Young, R. Wang, S. Sen, J. Kleinberg, A. Anderson: “Learning models of individual behavior in chess”, in Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp.1253–1263 (2022)
- [5] 稲川 ゆき： 「 『Skyrim VR』 ChatGPT で NPC と会話できる Mod デモ動画公開 — 「これ無しのプレイにはもう戻れない」 」 , <https://www.gamespark.jp/article/2023/04/28/129485.html>.
- [6] G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, A. Anandkumar: “Voyager: An open-ended embodied agent with large language models”, Transactions on Machine Learning Research (2024)