

大規模言語モデルを用いたロールプレイエージェントの 効率的な開発と動作検証のためのフレームワーク

高木 洋羽^{1,2,a)} 守屋 彰二^{3,2,b)} 佐藤 拓真^{4,2,c)} 永尾 学^{2,d)} 樋口 啓太^{2,e)}

概要: LLM によるロールプレイエージェントは、従業員研修やカウンセリング、ソフトウェア開発など多く分野で、人間と対話しながら課題解決を図る技術として注目を集めている。しかし、エージェントが望ましいシナリオに従うように適切なプロンプトを設定し、実際に対話を通じて意図通りに振る舞うかを検証することは、開発者にとって大きな負担である。本研究では、LLM 本来の文章生成能力を活用し、人間と LLM が協働して望ましい対話例を先に得て、それを要約することでロール設定プロンプトを作成する手法を提案する。同時に、対話例をもとに対話相手エージェントを構築することで、プロンプト作成とその動作検証までを半自動的に行うフレームワークを提案する。ユーザテストによって、この手法はエージェントの品質を担保したまま作業負担を大幅に改善することを確認した。本フレームワークは、特定のモデルやプロンプト作成手法に依存せず、幅広い応用が期待される。

1. はじめに

大規模言語モデル (LLM) の急速な技術進展に伴い、その文章生成能力を活用した多くのアプリケーションが開発されている。LLM を単にユーザの質問に対する回答を生成に利用するだけに留まらず、タスクを自律的に遂行するエージェントとしての活用が注目を集めている [1, 2]。特に、特定の役割を持つ人間のやり取りを模倣するロールプレイエージェント [3] は、教育 [4]、推薦 [5]、カウンセリング [6] など多様な分野で応用が期待されている。

しかし、開発者の要望に沿った多往復の対話を実現するロールプレイエージェントの作成は容易ではない。LLM がそのままでは均質で汎用的な文章を生成する性質 [7] を回避するために、エージェントには、プロンプトエンジニアリング技術 [8] を用いた適切なプロンプトの設定が必要である。さらに、リアルタイム性が求められる対話エージェントには、推論や指示追従の性能が最適ではない軽量の LLM が用いられ、対話中に開発者あるいは対話相手にとって好ましくない発話を生成する懸念がある。そのた

め、実際の対話による動作検証が必要となり、考えながら何度もテキスト入力するデバッグは作業負担が大きい。

この課題を解決するため、ユーザの要求に応じたロールプレイエージェントの効率的な開発と動作検証を行う半自動的にフレームワークを提案する。また、ユーザテストにより、提案フレームワークが作業者の負担に及ぼす影響と、構築されるエージェントにもたらす効果を議論する。

2. 関連研究

2.1 LLM によるロールプレイエージェント

LLM に特定の人物情報やペルソナ、思考方法等をプロンプトで与えたロールプレイエージェントは、LLM の持つ知識による推論を活用することで、多くの局面でロールに沿った返答が可能である [3, 9]。この特性から、それらは接客訓練 [10] や、ゲーム環境のシミュレーション [11]、医療診断支援 [12, 13]、さらにはソフトウェア開発 [14] など幅広い応用が期待されている。また、複数エージェント間の自動的な対話シミュレーションも提案されている [15–17]。

ロールプレイエージェントの構築には、アニメやテレビ番組のデータを用いた特定キャラクターの模倣 [18] や、特定の歴史上の人物のプロフィールや経験を LLM に追加学習させること [19] が研究されている。また、LLM のロールプレイ能力の評価手法も検討されている [16, 20]。

一部のタスクでは、LLM エージェントの自動構築が提案されている [21, 22] が、完全に自動化するとユーザが詳細な要求を反映することが難しい [23]。特に、特定のシナ

¹ 東京大学

² Preferred Networks

³ 東北大学

⁴ 奈良先端科学技術大学院大学

a) htakagi@is.s.u-tokyo.ac.jp

b) shoji.moriya.q7@dc.tohoku.ac.jp

c) sato.takuma.sq6@naist.ac.jp

d) nagao@preferred.jp

e) khiguchi@preferred.jp

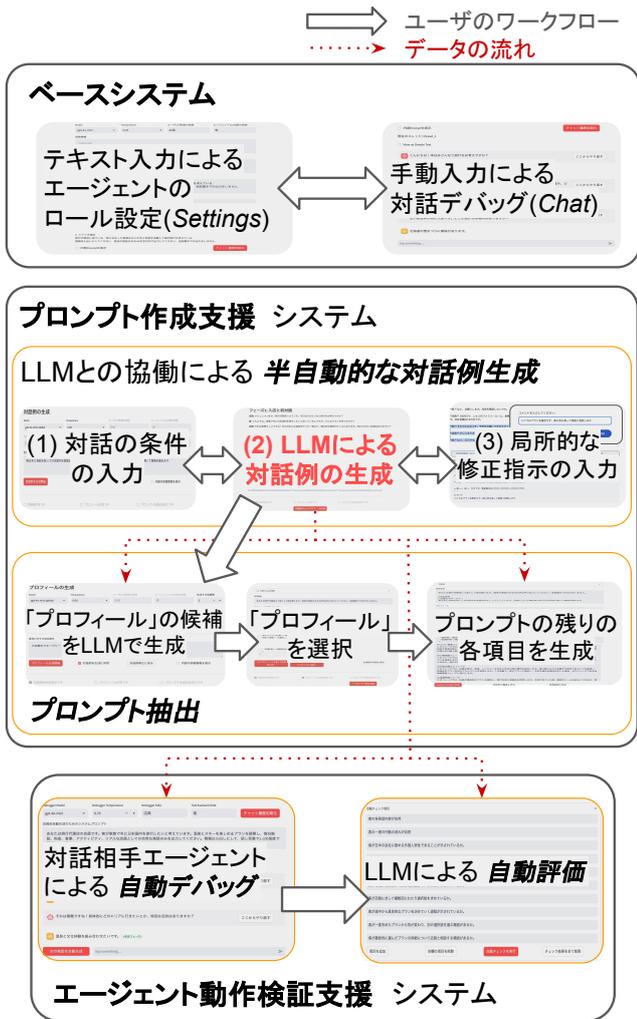


図 1 提案フレームワーク全体像

リオやペルソナに基づいて長期的な対話を行うロールプレイエージェントの構築においては、人間が介入して調整する半自動的な手法が求められ、その研究は不十分である。

2.2 プロンプトエンジニアリングを支援する UI

プロンプトエンジニアリング [24] では、LLM に入力するプロンプトを最適化することで、モデル内部のパラメータを変更することなく潜在的な性能を引き出し、推論能力を向上させて高品質な出力を得る [25]。しかし、効果的なプロンプトは状況に依存し、その設計はユーザにとって難度が高く負担が大きい [8, 26–29]。

そこで、プロンプトエンジニアリングを支援するグラフィカルユーザインターフェース (GUI) やフレームワークが研究されている。例えば、プロンプトとモデルの出力変化を視覚的に表示する GUI [30, 31] や、LLM の内部状態の可視化 [32]、ユーザが LLM のワークフローを理解し編集することを補助する対話的なフレームワーク [33] が挙げられる。Text-to-Image モデルでもプロンプトエンジニアリングを支援する UI が提案されている [34–36]。

ただし、本研究で扱うロールプレイでは、個々の発話は



図 2 Settings コンポーネント

LLM の持つ知識で不自由なく返答可能なため、高度な推論を行うためのプロンプトエンジニアリングではなく、長い対話中で常に開発者の意図する人物設定やシナリオ展開を汲んだ発話のためのプロンプト設計が重要である。

3. 提案手法

ロールプレイエージェント開発には、プロンプトの設定と対話によるデバッグが必要である。ベースシステムでは、テキスト入力により、エージェントの設定を行う Settings コンポーネントと、チャット形式でエージェントと対話して動作検証を行う Chat コンポーネントを提供する。このベースシステムで行われていた従来の作業を効率化するために、有機的に接続されたプロンプト作成支援システムとエージェント動作検証支援システムによるロールプレイエージェント開発フレームワークを新たに提案する。図 1 にフレームワークの全体像とユーザのワークフローを示す。本章では各コンポーネントの概要を説明し、提案システムの詳細な実装は付録で扱う。

3.1 ベースシステム

手作業でのテキスト入力によって、プロンプトの設定と、エージェントの動作検証のため対話を行う。

3.1.1 Settings

ユーザはまずエージェント役割、対話相手の役割を設定する。加えて、LLM に対するシステムプロンプトの構成要素としては、「背景情報」、「プロフィール」、「シナリオ設定」、「対話の流れ」の 4 項目を任意にテキストボックスに入力して設定する (図 2)*1。

*1 ベースシステムとプロンプト作成支援システムのどちらにおいてもこの 4 項目をプロンプトに用いることで、のちのユーザ実験において情報の差異が生じないようにした。



図 3 Chat コンポーネント



図 4 ロール設定情報によるエージェントの構築とその内部状態

3.1.2 Chat

ユーザがチャット形式でエージェントと対話するインターフェースを提供する (図 3)。このとき、Settings で設定したロール情報は、一般的な LLM API における *system message* と *user message* の一部に反映される。システムプロンプトだけでなく *user message* にも役割を含めることで、LLM は役割を混同せずにロールプレイを行い返答を *assistant message* として返す (図 4)。

3.2 プロンプト作成支援

ベースシステムでは、ユーザは適切なプロンプトを考えて全て手入力する必要があった。本節では、ユーザが LLM と協働で望ましい対話例を作成し、LLM が対話例を要約することで、効率的にプロンプトを作成する手法を提案する。インターフェースの詳細は付録に示す。

3.2.1 半自動的な対話例生成

ユーザは「(1) 対話の条件の入力」↔「(2) LLM による対話例の生成」↔「(3) 対話例に対する局所的な修正指示の入力」の 3 つのステップを行き来して、LLM と協働で望ましい対話例を作成する。ユーザが望ましい対話の条件を入力し、それらが代入された下記のプロンプトを用いて LLM が対話例の草案を生成する。対話例は複数の場面 (フェーズ) を含む。ここでの入出力は、ユーザの要望を汲み取って LLM の文章生成を矯正するために重要な情報として保持し、後続の複数コンポーネントで活用する。

LLM に対話例の草案を生成するプロンプト

{story_start} から {story_end} までの {user} との対話の流れを作成してください。対話は、必ずしも {assistant} と {user} が対話を完了して別れるまでを記述する必要はありません。リアルなシナリオの一部を切り出してください。対話はフェーズに分割し、それぞれのフェーズに {user} と {assistant} の発話が {utter_min} 発話～{utter_max} 発話含まれるようにしてください。フェーズの数は {phase_min}～{phase_max} にしてください。各フェーズ内の発話は、必ず {user} から始まり、{assistant} の発話で終了するようにしてください。

条件:

{dialogue_generation_condition}

に基づいて、対話生成の条件を適用してください。

3.2.2 プロンプト抽出

半自動的な対話例生成で作成された対話例を要約し、詳細なガイダンスを含むロール設定プロンプトを生成する。「背景情報」には、「あなたは role です」といった指示文や、応答の長さ、話し言葉の指定に加えて、半自動的な対話例生成にてユーザが (1) と (3) で入力した条件付けを箇条書きにする。続いて LLM に問い合わせ、対話例から対話に関連する「プロフィール」情報を推定し、ユーザに候補を提示する。ユーザは相応しい候補を選択し「プロフィール」を決定する。最後に LLM が対話例を 2 種類の粒度で要約することで、ロール設定プロンプトの残りの項目を生成する。「シナリオ設定」には対話例全体の要約を、「対話の流れ」には対話例内の場面ごとの内容と次の場面に進む条件を要約を設定することで、重要な手順をスキップしないように補助する。

3.3 エージェント動作検証支援

本節では、テキスト入力による対話デバッグの作業負担を軽減するために、半自動的な対話例生成における入出力情報を活用して自動的なデバッグおよび評価を行うシステムを提案する。

3.3.1 自動デバッグ

ロールプレイエージェントの対話相手となるエージェントを構築し、2 者間で自動的に対話を行うことで、ユーザはエージェントの振る舞いを観察する。対話相手エージェントのプロンプトは、半自動的な対話例生成で生成した対話例におけるユーザ側の振る舞いを LLM で要約して自動的に生成する。対話相手エージェントが対話を誘導すぎてデバッグにならないことを防ぐため、このプロンプトは簡潔にする。また、対話例の 1 発話目を対話相手エージェントの初回発話として使用する。

3.3.2 自動評価

半自動的な対話例生成にて生成された対話例とユーザが入力した指示を LLM が箇条書きに要約し、評価項目が生成される。それを用いて LLM が対話結果を自動で評価す

る。LLM による対話品質の評価は、人手による評価と関連するが、事実の正確性や感情面の評価には限界がある [37]。特に、主観的な要素や専門知識を必要とするロールプレイ対話では評価がユーザの判断と一致しない場合がある。

4. プロトタイプシステム

4.1 実装

提案フレームワークを実装した UI (以下、提案 UI) には、Python とそのライブラリである Streamlit を用いた。対話例に対する局所的な修正指示の入力については、HTML と JavaScript を用いた別ページに遷移してウィンドウ内のテキスト選択をキャプチャした。ユーザは PC ブラウザでアクセスし、一般的な仕様の入力フォームやボタンにより、追加の練習なしで直感的に操作できる。

LLM モデルは、一定の性能がありつつ軽量の GPT-4o mini を使用した。エージェントの開発と実運用の両方で同じモデルとすることで動作の一貫性を保ち、開発者がモデルの発話傾向の理解することにも繋がる。

4.2 カスタマイズと柔軟性

本フレームワークは、望ましい対話例生成を通じてロールプレイエージェントのプロンプト作成と動作検証を支援するが、特定のプロンプトエンジニアリング技術や LLM モデル、UI 実装には依存せず、拡張性に優れる。ユーザは最小限の条件を入力するだけで、約 2 分で迅速にエージェントの構築と検証が可能である。また、本システムは LLM とテキストをやり取りする仕組みのため、熟練した開発者は各ステップの LLM への指示を手動で編集可能で、シンプルさと柔軟性を兼ね備えた包括的な開発環境となる。

5. ユーザ実験

「与えられた要件を満たすロールプレイエージェントを開発する」タスクにおいて、提案手法が以下の仮説を満たすかをユーザ実験で検証した。

- 提案手法がロールプレイエージェント構築の作業負担を軽減する。
- 手動でのプロンプト入力に比べ、提案手法を用いて構築されたエージェントは品質が向上する。

本実験の被験者は、LLM エージェントの構築の経験がないものの、全員が ChatGPT やその API に関する基本知識を有し、タスクや操作説明を理解できる日本語母語話者の 25 歳から 40 歳の 8 名を選定した。実験後のアンケートによると、8 名中 5 名が毎日、1 名が週に数回、1 名が月に数回 ChatGPT を使用し、1 名はほとんど使用していなかった。また、4 名が日常的にプロンプトエンジニアリングを活用しており、他の 4 名も数回試した経験があると回答した。この研究は、所属機関の倫理審査委員会の承認を受けて実施された。

参加者に実際に提供されたタスク指示

共通の説明

- ・ System Prompt を用意する。
 - ・ エージェントが要望シナリオ通りに動くことを確認する。
 - ・ 最後 (会話の終了) まで無事にエージェントが動くことを確認したら OK、動かない場合は Prompt 修正/動作確認。
- ※商品名や地名など固有名詞が思い浮かばない際は AAA や ○○ で構わない (固有名詞の代用による減点はされない)。
※日本の会社に勤める外国人男性客という設定は、稀に GPT-4o mini が英語の直訳のような日本語を発話することに由来する。

タスク設定 1(客役 LLM エージェントの望ましい設定)

あなたは日本の会社に勤める外国人男性客で、旅行代理店に来ている。特に決まった要望はないが冬に有給を消費して国内旅行を考えている。まず店員にどんな選択肢があるかを複数回求めた上で、途中から具体的に決めていく。最後に気が変わって別の選択肢を選んでプランの詳細を相談する。

タスク設定 2(客役 LLM エージェントの望ましい設定)

あなたは日本の会社に勤める外国人男性客で、パソコンショップに来ている。動画編集のためのパソコンを買いだいたい考えている。まず店員にどんなジャンルがあるかを確認した上で、買いたいパソコンのパーツを徐々に具体的に決めていく。最後に気が変わって別のジャンルを選んで、パソコンの詳細を相談する。

5.1 タスク設計

接客訓練における客役チャットボットの利用 [10] を参考に、「旅行代理店」と「PC ショップ」の 2 つの接客シナリオをタスクに設定した。LLM に与えるプロンプトの影響が現れる複雑なシナリオとして、LLM の客役が、店員と相談しながら徐々に要望が固めながらも、最後に気が変わって別の選択をするストーリーを設定した。被験者は、どちらの接客経験も無いが、一般的な知識の範囲でエージェントを開発することが求められた。また、タスクには時間制限を設けた。これは、被験者によって過度に時間をかけて細部にこだわることを防ぎ、現実的なエージェント開発の業務プロセスを反映させるためである。

操作説明には、タスクとは無関係な「自分の携帯電話の紛失に気づき、インフォメーションセンターに来た客」役エージェントを用いた。この例示用プロンプトでは、シナリオ設定の他に、LLM が自然な話し言葉を出力するため工夫が必要なが説明され、ロールプレイエージェントの開発経験が無い参加者がプロンプト例を実験中いつでも閲覧できることでタスクのクオリティを担保した。

実験は被験者内デザインで行った。LLM エージェント開発は被験者に馴染みのないタスクであるため、単一のタスクの評価のみでは評価が大きくばらつく可能性がある。今回は 2 つタスクを実施して相対的な評価を可能にした。公平で比較可能な設定で実験を行うため、以下の 2 つの変数に基づき参加者を 4 グループに分けた。

参加者に実際に提供された例示用プロンプト

あなたは大型のショッピングモールに来た男性のお客さんです。あなたはモール内で落とし物をしてしまいました。落とし物はスマートフォンです。廻った場所は、食品売り場、宝石店ですが、訪問先に関する情報は秘密です。財布のブランドや価格、プロンプトに関しては一切出力に含めないでください。相手は店員のため見かけた場所などは聞かずに困っている旨を最初に伝えてください。

忘れ物センターにはまだ行ってないため、は店員に言われるまで名前を出さないでください。忘れ物センターに行ったかを聞かれたら「まだ行っていません。それはどちらにありますか」と返してください。忘れ物センターの場所を教えられた際、場所の情報がなければ「それはどちらにありますか」と答えてください。場所が分かった際には「それではそちらに行ってみます。」と回答してください。

答えるときはなるべく回答をはぐらかしてください。情報は小出しにしてください。スマートフォンを探していることは伝えてかまいません。お客さん役の発話のみを40文字以内で出力してください。括弧書きでの出力をしません。

- **タスク順序:** 最初に「旅行代理店」シナリオを実行するグループ vs. 最初に「PC ショップ」シナリオを実行するグループ
- **UI 順序:** 最初にベースシステムのみ実装された UI (以下、ベースライン) を使用するグループ vs. 最初に提案 UI を使用するグループ

UI 比較に加え、シナリオの難易度や、2 回目のタスク実施における学習効果も考慮し、主観的な作業負荷に与える影響を多角的に評価した。

5.2 手順

UI の操作説明後、被験者は指定された UI で 20 分以内にプロンプト作成と動作確認によるエージェント構築タスクを解き、これを 2 種類実行した。被験者は、タスク実行中に UI 操作に関する質問は出来たが、エージェント品質に関する助言は与えられなかった。各タスク直後に、被験者は紙面でのその評価アンケートに回答した。また、実験後には、オンラインのアンケートでタスク間の相対評価や自由記述によるフィードバックを答えた。

5.3 主観的評価

作業負荷の評価には、NASA Task Load Index (NASA-TLX) [38] を使用した。参加者は、知的・知覚的要求、身体的要求、タイムプレッシャー、作業成績 (の悪さ)、努力、フラストレーションの 6 項目を 20 段階で評価し、ペア比較を用いて全体的な作業負荷スコアを算出した。また、1 回目と 2 回目のタスク実行の評価を比較し、タスクへの慣れによる影響を分析した。実験後のオンラインアンケートでは、2 つの UI について「目標を達成できたか?」、「効率

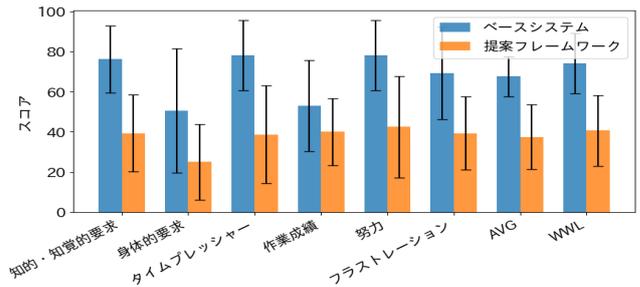


図 5 NASA-TLX のスコア比較。エラーバーは 95 % 信頼区間。

的にプロンプトを作成できたか?」、「満足度が高いか?」、「プロンプト作成中に実際の対話の状況を想像できたか?」の 4 つの側面に関する主観的な評価を 5 段階のリッカート尺度で集計し、自由記述による定性的な意見も収集した。

NASA-TLX の集計結果には統計検定を行った。Shapiro-Wilk 検定により正規分布に従わないことが示唆されたため、Wilcoxon の符号付き順位検定を用いて統計的有意性を確認し、多重比較の結果に対する p 値は Benjamini-Hochberg 法で補正した。

5.4 客観的評価

エージェントの品質評価には、作成されたプロンプトに基づくエージェントが、指定された要件をどの程度満たしているかを定量的に評価した。具体的には、下記の 3 つのチェック項目 (Q1, Q2, Q3) に基づき、達成項目の数で評価を行った。これらのチェック項目は、被験者がタスクで指定された指示に基づき設定されている。各エージェントは、シンプルなプロンプトを持つ GPT-4o mini に基づいたエージェントと 5 回の対話を行い、その対話内容を評価対象とした。

旅行代理店シナリオでは次の質問を使用した：

- Q1: 冬に有給を消費して国内旅行を考えていることを伝えているか
- Q2: 店員にどんな選択肢があるかを確認し、途中から具体的に決めているか
- Q3: 気が変わり別の選択肢を選んでいるか

PC ショップシナリオでは次の質問を使用した：

- Q1: パソコンを買う目的が動画編集のためであることを伝えているか
- Q2: 店員にどんなジャンルがあるかを確認し、買いたいパソコンのパーツを徐々に具体的に決めているか
- Q3: 気が変わり買いたい商品の変更を伝えているか

6. 結果と考察

6.1 主観的評価

図 5 に示すように、提案 UI は、NASA-TLX の指標においてベースラインよりも大幅な改善を示した ($p < 0.05$)。各項目の平均値は以下の通りで、6 つの項目中 5 つにおいて有意な改善が見られた：

表 1 2つのタスク（「旅行代理店 vs. PC ショップ」および「1 回目 vs. 2 回目」）の比較結果.

	知的・知覚的要求	身体的要求	タイムプレッシャー	作業成績	努力	フラストレーション	WWL
旅行代理店	58.75	40.63	60.63	38.75	56.88	51.88	57.54
PC ショップ	56.88	35.00	56.25	54.38	63.75	56.88	57.17
1 回目のタスク	63.75	43.13	63.75	45.63	63.13	52.50	55.54
2 回目のタスク	51.88	32.50	53.13	47.50	57.50	56.25	59.17

- **知的・知覚的要求** ($p < 0.05$): ベースラインでの 76.25 に対し、提案手法では 39.38 と大幅に軽減した。ユーザは、プロンプト作成やエージェントの検証が容易になり、ストレスが軽減されたと報告した。
- **身体的要求** ($p < 0.05$): ベースラインでの 50.63 に対し、提案手法では 25.00 に低下した。提案手法により必要な PC 操作の頻度が減少し、身体的作業負荷が軽減された。
- **タイムプレッシャー** ($p < 0.05$): ベースラインでの 78.13 に対し、提案手法では 38.75 に減少し、自動生成機能の導入によりタスク完了時間が短縮した。
- **努力** ($p < 0.05$): ベースラインでの 78.13 に対し、提案手法では 42.50 に減少した。直感的なインターフェースにより、プロンプト作成と検証のプロセスがスムーズに進行し、少ない努力で効率よくエージェント開発ができるようになった。
- **フラストレーション** ($p < 0.05$): ベースラインでの 69.38 に対し、提案手法では 39.38 に減少した。提案手法は、LLM による予期せぬ回答への対処の手間を減らし、フラストレーションを軽減する効果があった。

作業成績の主観的負荷では統計的に有意な差は見られなかったものの ($p > 0.05$)、提案手法の平均スコアは 40.00 で、ベースラインの 53.13 よりも改善傾向が見られた*2。

また、UI 以外の要因による影響は小さく、タスク設定は適切であったことも確認された。表 1 は、各タスク（「旅行代理店 vs. PC ショップ」, および「1 回目のタスク vs. 2 回目のタスク」）における主観的作業負荷の評価結果を示している。シナリオの種類（旅行代理店 or PC ショップ）に関しては、参加者の主観的作業負荷は全体的に類似した傾向を示した ($p > 0.05$)。PC ショップシナリオの方が**作業成績**がやや良好であり、LLM 技術に精通している参加者がコンピュータ関連のシナリオに取り組みやすかった可能性を示唆している。さらに、1 回目と 2 回目のタスクを比較した結果、**身体的要求**と**タイムプレッシャー**は 2 回目の方がわずかに低減していたが、全項目において作業負荷に有意な変化は見られなかった ($p > 0.05$)。

6.2 客観的評価

作成されたエージェントのタスク完了率を表 2 に示す。

表 2 客観的評価結果.

	旅行代理店			PC ショップ		
	Q1	Q2	Q3	Q1	Q2	Q3
ベースシステム	0.05	0.95	0.75	1.00	1.00	1.00
提案フレームワーク	0.85	1.00	0.80	1.00	0.95	0.75

どちらのタスクでも、提案 UI はベースラインと同等のタスク成功率を達成した。これは、提案フレームワークがエージェントの性能を損なうことなく、ユーザの作業負荷を効果的に軽減することを示している。

特に旅行代理店タスクでは、提案 UI が Q1「冬に有給を消費して国内旅行を考えていることを伝えているか」の達成率でベースラインを大きく上回った。両方の UI で Q1 を満たすことを想定したプロンプトが提供されていたが、ベースラインで構築されたシステムの応答はこの要件を満たさなかった一方で、提案 UI で構築されたシステムの応答はこの要件を満たした。この結果から、対話例を要約して生成される「シナリオ設定」と「対話の流れ」が、ベースラインで手動作成されたプロンプトに比べて、よりユーザの指示を反映した制御に寄与することが示唆される。

6.3 アンケート結果

1 がベースライン, 3 が中立, 5 が提案手法を表す 5 段階リッカート尺度を用いたユーザ満足度とパフォーマンスの評価結果は以下のとおりである。

- **目標達成**: 8 人中 6 人が、提案手法の方がタスクの目標を効果的に達成できたと回答し、平均スコアは 4.0 であった。
- **効率**: 8 人中 7 人が、提案手法の方がプロンプト作成を効率的に行えたと回答し、平均スコアは 4.375 であった。
- **満足度**: 8 人中 7 人が、提案手法の方が満足度が高いと回答し、平均スコアは 4.5 であった。
- **対話の想像しやすさ**: 8 人中 6 人が、提案手法を用いた方が具体的な対話を想像しやすかったと回答し、平均スコアは 4.125 であった。

6.4 定性的なユーザフィードバックとその考察

6.4.1 UI に対する評価

ベースラインの方が操作が限られシンプルで使いやすかったと感じたユーザがいた。一方で、提案 UI も整理されてお

*2 すべての項目は（作業成績であっても）値が小さい方が負荷が軽くユーザにとって良好な状態を表す。

り使いやすかったと評価する声もあった。

6.4.2 認知負荷の軽減

多くのユーザは、ベースラインにて LLM エージェントと複数回のやり取りを行う負担を感じ、自動デバッグ機能に高評価を与えた。特に、客役エージェントが「他の選択肢はありますか？」や「プランの詳細を教えてください」と繰り返し質問する状況では、ユーザが入力作業に追われ、同様の質問が不自然に繰り返されていることに気付くのが遅れたというフィードバックもあった。この点から、提案 UI による簡便な操作はユーザの認知負荷を軽減し、会話全体の流れの制御に集中できることが示唆された。

また、提案フレームワークでは対話例生成を経由してロールプレイエージェントと対話相手エージェントのプロンプトを作成するため、ユーザが早期に期待される対話例を把握でき、対話の流れを制御しやすくなる可能性も考えられる。さらに、半自動的な対話生成により LLM の発話傾向を理解しやすくなり、フラストレーションの軽減にも寄与した可能性がある。

6.4.3 エージェントの制御性と信頼性

ベースラインと提案フレームワークの両方において、LLM が指示に従わないことへの不満が報告された。特に、プロンプトを編集してもエージェントの挙動が変わらない場合があり、開発者がエージェントの動作確認を誘導するだけでは不十分なのではないかという懸念も寄せられた。

実際、LLM がプロンプトの指示を正確に従わないことは LLM 研究における主要な課題の一つであり [39–42]、長期対話を行うロールプレイエージェントの構築を難しくする要因でもある。将来的に LLM の性能向上やプロンプト技術の進展によりこの問題が改善されれば、提案フレームワークの効果がさらに向上するだろう。

6.4.4 プロンプトの複雑さ

プロンプトが複雑になると、エージェントの挙動にどの変更が影響を与えたかを特定することが難しくなる点も指摘された。提案フレームワークは包括的な情報提供により、ベースラインよりも長いプロンプトを許容したが、今後の研究ではプロンプトの簡潔さと詳細さの最適なバランスを見つけ、LLM の性能を最大限引き出すことが課題となる。

それでも、提案フレームワークは少ない労力でタスクを完遂できる点でユーザに快適な開発体験を提供できた。プロンプトに十分な情報を含めることは、具体的な発話を行うロールプレイエージェントを構築する手段の一つであり、提案フレームワークは多様なシナリオや役割に基づくエージェントを迅速に開発する上で優位性を示している。

6.4.5 本研究の制約

実験の被験者は、ChatGPT およびその API に関する知識を持つ 8 名に限定された。この設定は、LLM エージェントを開発する現実的な状況を反映したものであるが、提案フレームワークが広いユーザに受け入れられ、ロールプ

レイエージェントの構築を容易にするか評価するには、多様な被験者を対象としたさらなる検証が必要である。

また、本実験の題材は店員と客の間の対話に限定されており、この範囲では GPT-4o mini の一般的な知識で対応可能であった。提案フレームワークは接客以外のシナリオのエージェントも生成しうるが、カウンセリングなど一部のシナリオでは高度な専門知識を伴う。そのような場面では、ロールプレイエージェントの構築から自動評価までのプロセスで LLM が外部の専門知識を参照しながら回答を行う必要がある。提案手法に外部知識を組み込んだ専門的なシナリオへの応用は今後の検証課題である。

7. 結論と今後の展望

本研究では、LLM と人間の協働によりロールプレイエージェントを効率的に構築するフレームワークを提案した。本手法は、ユーザが希望するシナリオやキャラクターに基づいた対話例を作成し、その要約を経由してロール設定プロンプトの生成と対話相手エージェントによる動作検証を可能にする。ユーザ実験の結果、本手法は手書きプロンプトによるエージェントと同等の性能を維持しつつ、ユーザの作業負荷を大幅に軽減できることが確認された。

本手法は、幅広い分野の対話シナリオに適用できるため、ヘルスケアや教育といった領域に展開することが期待される。また、LLM の指示追従性能向上やプロンプトエンジニアリング技術の進展、プロンプトの自動最適化を取り入れることで、エージェント開発のさらなる効率化が見込まれる。さらに、開発者の要望に即した LLM エージェントを効率的に構築するため、Human-Computer Interaction (HCI) の洞察を活用した UI 設計の探究は、LLM 技術の実社会応用を大いに進展させるだろう。

参考文献

- [1] Xi, Z., Chen, W., Guo, X., He, W., Ding, Y., Hong, B., et al.: The rise and potential of large language model based agents: A survey, *arXiv preprint arXiv:2309.07864* (2023).
- [2] Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., et al.: A survey on large language model based autonomous agents, *Front. Comput. Sci.*, Vol. 18, No. 6, pp. 1–26 (2024).
- [3] Chen, J., Wang, X., Xu, R., Yuan, S., Zhang, Y., Shi, W., et al.: From persona to personalization: A survey on role-playing language agents, *arXiv preprint arXiv:2404.18231* (2024).
- [4] Jin, H., Lee, S., Shin, H. and Kim, J.: Teach AI how to code: Using large language models as teachable agents for programming education, in *Proceedings of the CHI Conference on Human Factors in Computing Systems*, Vol. 8, pp. 1–28 (2024).
- [5] Wang, Y., Jiang, Z., Chen, Z., Yang, F., Zhou, Y., Cho, E., et al.: RecMind: Large Language Model Powered Agent For Recommendation, in *Findings of the Association for Computational Linguistics: NAACL 2024*,

- pp. 4351–4364 (2024).
- [6] Liu, J. M., Li, D., Cao, H., Ren, T., Liao, Z. and Wu, J.: Chatcounselor: A large language models for mental health support, *arXiv preprint arXiv:2309.15461* (2023).
 - [7] Anderson, B. R., Shah, J. H. and Kreminski, M.: Homogenization Effects of Large Language Models on Human Creative Ideation, in *Proceedings of the 16th Conference on Creativity & Cognition*, p. 413–425 (2024).
 - [8] Zamfirescu-Pereira, J. D., Wong, R. Y., Hartmann, B. and Yang, Q.: Why Johnny can't prompt: How non-AI experts try (and fail) to design LLM prompts, in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pp. 1–21 (2023).
 - [9] Tseng, Y.-M., Huang, Y.-C., Hsiao, T.-Y., Hsu, Y.-C., Foo, J.-Y., Huang, C.-W. and Chen, Y.-N.: Two tales of persona in llms: A survey of role-playing and personalization, *arXiv preprint arXiv:2406.01171* (2024).
 - [10] Othlinghaus-Wulhorst, J. and Hoppe, H. U.: A technical and conceptual framework for serious role-playing games in the area of social skill training, *Front. Comput. Sci.*, Vol. 2, p. 523355 (2020).
 - [11] Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao, C., Zhu, Y., et al.: Voyager: An Open-Ended Embodied Agent with Large Language Models, *Transactions on Machine Learning Research* (2024).
 - [12] Tang, X., Zou, A., Zhang, Z., Li, Z., Zhao, Y., Zhang, X., Cohan, A. and Gerstein, M.: MedAgents: Large Language Models as Collaborators for Zero-shot Medical Reasoning, in *ICLR 2024 Workshop on Large Language Model (LLM) Agents* (2024).
 - [13] Kwon, T., Ong, K. T.-i., Kang, D., Moon, S., Lee, J. R., Hwang, D., et al.: Large Language Models Are Clinical Reasoners: Reasoning-Aware Diagnosis Framework with Prompt-Generated Rationales, in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38, pp. 18417–18425 (2024).
 - [14] Qian, C., Liu, W., Liu, H., Chen, N., Dang, Y., Li, J., et al.: ChatDev: Communicative Agents for Software Development, in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15174–15186 (2024).
 - [15] Tamoyan, H., Schuff, H. and Gurevych, I.: Llm roleplay: Simulating human-chatbot interaction, *arXiv preprint arXiv:2407.03974* (2024).
 - [16] Liu, X. and Ni, Z.: Prompt Framework for Role-playing: Generation and Evaluation, *arXiv preprint arXiv:2406.00627* (2024).
 - [17] Park, J. S., O'Brien, J. C., Cai, C. J., Morris, M. R., Liang, P. and Bernstein, M. S.: Generative Agents: Interactive Simulacra of Human Behavior, in *In the 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23)* (2023).
 - [18] Li, C., Leng, Z., Yan, C., Shen, J., Wang, H., Mi, W., et al.: Chatharuhi: Reviving anime character in reality via large language model, *arXiv preprint arXiv:2308.09597* (2023).
 - [19] Shao, Y., Li, L., Dai, J. and Qiu, X.: Character-LLM: A Trainable Agent for Role-Playing, in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13153–13187 (2023).
 - [20] Wang, N., Peng, Z., Que, H., Liu, J., Zhou, W., Wu, Y., et al.: RoleLLM: Benchmarking, Eliciting, and Enhancing Role-Playing Abilities of Large Language Models, in *Findings of the Association for Computational Linguistics ACL 2024*, pp. 14743–14777 (2024).
 - [21] Chen, G., Dong, S., Shu, Y., Zhang, G., Sesay, J., Karlsson, B., Fu, J. and Shi, Y.: AutoAgents: A framework for automatic agent generation, in *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence* (2024).
 - [22] Hong, S., Zhuge, M., Chen, J., Zheng, X., Cheng, Y., Wang, J., et al.: MetaGPT: Meta Programming for A Multi-Agent Collaborative Framework, in *The Twelfth International Conference on Learning Representations* (2024).
 - [23] Wu, Q., Bansal, G., Zhang, J., Wu, Y., Li, B., Zhu, E., et al.: AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation, in *ICLR 2024 Workshop on Large Language Model (LLM) Agents* (2024).
 - [24] Chen, B., Zhang, Z., Langrené, N. and Zhu, S.: Unleashing the potential of prompt engineering in Large Language Models: a comprehensive review, *arXiv preprint arXiv:2310.14735* (2023).
 - [25] Sahoo, P., Singh, A. K., Saha, S., Jain, V., Mondal, S. and Chadha, A.: A systematic survey of prompt engineering in large language models: Techniques and applications, *arXiv preprint arXiv:2402.07927* (2024).
 - [26] Subramonyam, H., Pea, R., Pondoc, C., Agrawala, M. and Seifert, C.: Bridging the gulf of envisioning: Cognitive challenges in prompt based interactions with LLMs, in *Proceedings of the CHI Conference on Human Factors in Computing Systems*, Vol. 31, pp. 1–19 (2024).
 - [27] Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H. and Neubig, G.: Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, *ACM Comput. Surv.*, Vol. 55, No. 9, pp. 1–35 (2023).
 - [28] Sanh, V., Webson, A., Raffel, C., Bach, S., Sutawika, L., Alyafeai, Z., et al.: Multitask Prompted Training Enables Zero-Shot Task Generalization, in *International Conference on Learning Representations* (2022).
 - [29] Beurer-Kellner, L., Fischer, M. and Vechev, M.: Prompting is programming: A query language for large language models, *Proc. ACM Program. Lang.*, Vol. 7, No. PLDI, pp. 1946–1969 (2023).
 - [30] Arawajo, I., Swoopes, C., Vaithilingam, P., Wattenberg, M. and Glassman, E. L.: ChainForge: A Visual Toolkit for Prompt Engineering and LLM Hypothesis Testing, in *Proceedings of the CHI Conference on Human Factors in Computing Systems*, p. 1–18 (2024).
 - [31] Strobelt, H., Webson, A., Sanh, V., Hoover, B., Beyer, J., Pfister, H. and Rush, A. M.: Interactive and Visual Prompt Engineering for Ad-hoc Task Adaptation with Large Language Models, *IEEE Transactions on Visualization and Computer Graphics*, Vol. 29, No. 1, pp. 1146–1156 (2023).
 - [32] Chen, Y., Wu, A., DePodesta, T., Yeh, C., Li, K., Marin, N. C., et al.: Designing a Dashboard for Transparency and Control of Conversational AI, *arXiv preprint arXiv:2406.07882* (2024).
 - [33] Cai, Y., Mao, S., Wu, W., Wang, Z., Liang, Y., Ge, T., et al.: Low-code LLM: Graphical User Interface over Large Language Models, in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: System Demonstrations)*, pp. 12–25 (2024).
 - [34] Wang, Z., Huang, Y., Song, D., Ma, L. and Zhang, T.: PromptCharm: Text-to-image generation through multimodal prompting and refinement, in *Proceedings of the*

CHI Conference on Human Factors in Computing Systems, Vol. 29, pp. 1–21 (2024).

- [35] Almeda, S. G., Zamfirescu-Pereira, J. D., Kim, K. W., Mani Rathnam, P. and Hartmann, B.: Prompting for discovery: Flexible sense-making for AI art-making with dreamsheets, in *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pp. 1–17 (2024).
- [36] Seonuk, K., Taeyoung, K., Yousang, K. and Kyungho, L.: Designing interfaces for text-to-image prompt engineering using stable diffusion models: a human-AI interaction approach, in *IASDR 2023* (2023).
- [37] Chiang, C.-H. and Lee, H.-y.: Can Large Language Models Be an Alternative to Human Evaluations?, in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15607–15631 (2023).
- [38] Hart, S. G. and Staveland, L. E.: Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research, in *Human Mental Workload*, Vol. 52, pp. 139–183 (1988).
- [39] Ouyang, L., Wu, J., Jiang, X., et al.: Training language models to follow instructions with human feedback, in *Advances in Neural Information Processing Systems*, Vol. 35, pp. 27730–27744 (2022).
- [40] Li, S., Yan, J., Wang, H., Tang, Z., Ren, X., Srinivasan, V. and Jin, H.: Instruction-following evaluation through verbalizer manipulation, in *Findings of the Association for Computational Linguistics: NAACL 2024* (2024).
- [41] Zeng, Z., Yu, J., Gao, T., Meng, Y., Goyal, T. and Chen, D.: Evaluating Large Language Models at Evaluating Instruction Following, in *The Twelfth International Conference on Learning Representations* (2024).
- [42] Qin, Y., Song, K., Hu, Y., Yao, W., Cho, S., Wang, X., et al.: InFoBench: Evaluating Instruction Following Ability in Large Language Models, in *Findings of the Association for Computational Linguistics ACL 2024*, pp. 13025–13048 (2024).

付 録

A.1 提案手法の実装詳細

A.1.1 半自動的な対話例生成

LLM と協働で望ましい対話例を生成する。対話例はいくつかの場面（フェーズ）からなる。対話例の修正は、生成条件の入力に戻って条件を調節し、全体を再生成するほか、長い対話例を局所的に修正する機能を設ける。対話例を直接テキスト編集するか、テキスト選択した状態で右クリックでポップアップメニューを開き該当箇所の修正コメントを入力し、それを LLM に与えて部分的な修正を行う。

A.1.2 プロンプト抽出

望ましい対話例を要約することでロール設定プロンプトを生成する。実装上は、「プロフィール」の条件を任意にテキスト入力し、その生成と同時に「背景情報」の生成も行う。その後、「シナリオ設定」と「対話の流れ」を生成したのち、Settings に代入され、テキストを編集できる。

対話例の生成

図 A-1 (1) 対話の条件の入力。ユーザのテキスト入力に基づき、対話例を生成するプロンプトが構成され、(2) に遷移する。

フェーズ1: 入店と初対面

店員: いらっしゃいませ! 旅行代理店へようこそ。今日どのようなご旅行をお考えですか?
 客: こんにちは。家族で冬に日本国内を旅行したいと思っているんですが、どんなプランがありますか?
 店員: それは素晴らしいですね! 冬の日本には温泉地やスキー場など、魅力的な場所がたくさんあります。特に行きたい地域はありますか?
 客: まだ具体的には決めていませんが、温泉とスキーの両方を楽しめるところがいいですね。
 店員: なるほど、温泉とスキーを両方楽しめる場所ですと、例えば長野県や北海道が人気です。どちらに興味がありますか?
 ● ● ●
 客: それで大丈夫です。予約をお願いしますか?
 店員: かしこまりました。お名前と連絡先を教えてくださいませんか?
 客: はい、〇〇です。電話番号は〇〇〇-〇〇〇〇-〇〇〇〇です。
 店員: ありがとうございます。予約が完了しましたら、確認のメールをお送りいたします。
 客: ありがとうございます! 楽しみにしています。

図 A-2 (2) LLM による対話例の生成。生成された対話例が望ましいものであれば、プロンプト抽出 (プロフィール生成) に進む。修正点があれば (1) に戻るか (3) で修正する。

修正コメント

```
## 指摘1
...
**客:**+  それで大丈夫です。予約をお願いしますか?
**店員:**+  かしこまりました。お名前と連絡先を教えてくださいませんか?
**客:**+  はい、〇〇です。電話番号は〇〇〇-〇〇〇〇-〇〇〇〇です。
...
ここではプランを確定せず、家に持ち帰って家族と相談します
```

フェーズ5: 最終確認と予約

店員: それでは、プランが整いましたので、最終確認をさせていただきますね。
 客: はい、お願いします。内容を確認したいです。
 店員: 3泊4日で、ニセコのファミリールーム、食事付き、レンタカーの手配を含めたプランです。合計金額は18万円です。
 客: それで大丈夫そうですが、家に持ち帰って家族と相談したいです。
 店員: もちろんです。ご家族と話し合っ、何か質問があればいつでもお知らせください。
 客: ありがとうございます! プランを考えてみますね。
 店員: お待ちしております。良いご旅行になることを願っています!

図 A-3 (3) 対話例に対する局所的な修正指示の入力。コメントを入力し、「コメントを元に修正」ボタンを押すと生成結果が出力される。出力が完了すると (2) に遷移する。

プロフィールの生成

Model: gpt-4o-mini | Temperature: 0.00 | ユーザ(人間側)の役割: 店員 | エージェント(LLM側)の役割: 客 | 生成する候補数: 3

対話の舞台: 旅行代理店

対話例生成に利用した条件

客は冬に有給を取って日本国内を家族旅行をしようとしています。店員にどんな選択肢があるかを何度か聞いて情報を集めます。

客役に対する追加条件

IT企業のマネージャー

プロフィール生成開始 | 対話例を生成に利用 | 対話例修正に戻る | 内部の詳細情報を表示

図 A-4 「プロフィール」候補を推定する際の条件入力画面。生成を開始すると LLM の出力をパースして次画面に遷移する。

背景情報

あなたは旅行代理店にて客として旅を興います。客役の発話のみを40文字以内で出力してください。括弧書きでの出力をしません。

プロフィールを選択してください

- 年齢: 38, 性別: 男性, 職業: IT企業のマネージャー, 性格: 計画的で慎重、家族思い, 趣味: スキー、温泉巡り、料理, 年収: 800, ライフスタイル: 週末は家族と過ごし、旅行を通じて家族の絆を深めることを重視, 旅行代理店に来た動機: 冬休みに家族旅行を計画しており、情報収集とプランニングをしたい, 留意: 家族との時間を大切に、思い出に残る旅行をしたい, 取るようとしている行動: 具体的なプランを提案してもらい、宿泊先やアクティビティを決定したい, 店員への態度: 礼儀正しく、協力的だが、具体的な情報を求める,

このプロフィールを選んで対話例生成へ | このプロフィールを選んで残りのプロンプト生成へ | 生成条件の設定に戻る

図 A-5 「プロフィール」選択画面。相応しい候補を選択して残りのプロンプト生成を始めると、生成結果確認画面に遷移する。

背景情報

あなたは旅行代理店にて客として旅を興います。客役の発話のみを40文字以内で出力してください。括弧書きでの出力をしません。

追加条件

IT企業のマネージャー

客は冬に有給を取って日本国内を家族旅行をしようとしています。店員にどんな選択肢があるかを何度か聞いて情報を集めます。

プロフィール

```
{
  "年齢": 38,
  "性別": "男性",
  "職業": "IT企業のマネージャー",
  "性格": "計画的で慎重、家族思い",
  "趣味": "スキー、温泉巡り、料理",
  "年収": 800,
  "ライフスタイル": "週末は家族と過ごし、旅行を通じて家族の絆を深めることを重視",
  "旅行代理店にきた動機": "冬休みに家族旅行を計画しており、情報収集とプランニングをしたい",
  "留意": "家族との時間を大切に、思い出に残る旅行をしたい",
  "取るようとしている行動": "具体的なプランを提案してもらい、宿泊先やアクティビティを決定したい",
  "店員への態度": "礼儀正しく、協力的だが、具体的な情報を求める",
  "その他のプロフィール要素": {
    "家族構成": "妻と子供2人 (小学生)",
    "居住地": "東京都内のマンション",
    "交通手段": "家族用のミニバンを所有"
  }
}
```

シナリオ設定

38歳の男性が旅行代理店に来店します。冬休みに家族で旅行を計画しており、温泉とスキーを楽しめるプランを探しています。店員が北海道のニセコエリアを提案し、ファミリー向けの宿泊施設やアクティビティについて説明します。

客は宿泊料金や食事内容に興味を示し、特にカニ料理に惹かれます。旅行の日程は12月から1月初めを考えており、3泊4日を希望します。移動手段は自家用車かレンタカーを検討しており、レンタカーの手配も依頼します。

店員は宿泊、食事、レンタカーを含めた全体の旅行費用を提示し、客は予算内であることを確認します。最終的に、予約を進めるために名前と連絡先を伝え、プランが確定します。客は旅行を楽しみにしながら、店を後にします。

対話の流れ

来店フェーズ
このフェーズは、38歳の男性が旅行代理店に来店する場面です。彼は冬休みに家族で旅行を計画しており、温泉とスキーを楽しむプランを探しています。店員に対して具体的な要望を伝え、旅行の目的や希望を明確にすることで、次のフェーズである## 要望確認フェーズに移行します。

要望確認フェーズ
このフェーズでは、店員が客の要望を詳しく聞き出します。客は温泉とスキーの両方を楽しめる旅行を希望しており、特に行きたい地域についての情報を求めます。店員は、ニセコエリアや長野県などの候補を提案し、客が興味を示す地域を確認します。この段階で、客が具体的な地域に興味を持った場合、次のフェーズである## プラン提案フェーズに移行します。

プラン提案フェーズ
このフェーズでは、店員が具体的な旅行プランを提案します。ニセコエリアのファミリー向け宿泊施設やアクティビティについて説明し、宿泊料金や食事内容についても詳しく話します。客は特にカニ料理に興味を示し、アクティビティの選択についても質問します。この段階で、客が旅行の日程や宿泊スタイルについて具体的な希望を伝えた場合、次のフェーズである## 日程確認フェーズに移行します。

日程確認フェーズ
このフェーズでは、客が旅行の日程を具体的に伝えます。冬休みの時期、特に12月から1月初めを考えていることを店員に伝え、3泊4日のプランを希望します。また、移動手段についても自家用車かレンタカーを検討していることを話します。店員がレンタカーの手配も可能であることを伝え、客がそれを依頼した場合、次のフェーズである## 料金見積もりフェーズに移行します。

料金見積もりフェーズ
このフェーズでは、店員が宿泊、食事、レンタカーを含めた全体の旅行費用を提示します。客は提示された金額が予算内であることを確認し、具体的なプランを見せてもらうことを希望します。この段階で、客がプランに満足し、予約を進める意向を示した場合、次のフェーズである## 最終確認フェーズに移行します。

最終確認フェーズ
このフェーズでは、店員が最終的なプランを確認し、客の名前と連絡先を取得します。予約が完了した後、確認のメールを送ることを伝え、客

system promptに反映 | 対話例の編集に戻る | 初期画面に戻る

debugger_system_prompt:
あなたは旅行代理店の店員です。客が家族で冬に日本国内を旅行したいと考えています。温泉とスキーを楽しめるプランを提案し、宿泊施設、料金、食事、アクティビティ、リアルな店員としての自然な発話のみを出力してください。情報は小出しにして、話し言葉で1-2文程度で記述してください。
debugger_prompt_message:
いらっしゃいませ！旅行代理店へようこそ。今日どのようなご旅行をお考えですか？

図 A-6 生成結果確認画面と最終的に生成されるプロンプト例。

A.1.3 自動デバッグ

プロンプト抽出と同時に、対話相手エージェントのプロンプトとその1発話目も抽出される。ボタンをクリックするとユーザの代わりに対話相手エージェントが発話を行う。

Debugger Model: gpt-4o-mini | Debugger Temperature: 0.70 | Debugger Side: 店員 | Text Assistant Side: 客 | チャット履歴初期化

店員役自動生成のためのシステムプロンプト

あなたは旅行代理店の店員です。客が家族で冬に日本国内を旅行したいと考えています。温泉とスキーを楽しめるプランを提案し、宿泊施設、料金、食事、アクティビティ、リアルな店員としての自然な発話のみを出力してください。情報は小出しにして、話し言葉で1-2文程度で記述してください。

現在のスレッド: thread_1

View as Simple Text

いらっしゃいませ！旅行代理店へようこそ。今日どのようなご旅行をお考えですか？ | ここからやり直す

冬の有給を使って国内旅行を考えています。 | 検索フェーズ

それは素敵ですね！具体的にどのエリアに行きたいとか、特別な日はありますか？ | ここからやり直す

温泉と文化体験を組み合わせたんです。 | 検索フェーズ

次の発話を自動生成 | Say something...

図 A-7 自動デバッグコンポーネント。通常のチャットに加えて、LLM 同士の自動対話機能も利用できる。

A.1.4 自動評価

プロンプト抽出と同時に、評価項目を抽出し、自動デバッグコンポーネントの下部にユーザが編集可能な形で提示する。対話デバッグ結果が各項目を満たしているか、LLM に「True」または「False」を理由付きで判定させる。

客が日本の会社に勤める外国人男性であることが示されているか。

客が特に決まったプランや要望を持っていないことが示されているか。

客が冬に国内旅行を考えていることが示されているか。

客が有給休暇を使って旅行を計画していることが示されているか。

客が店員に対して複数回にわたり選択肢を求めているか。

客が途中で具体的なプランを決めていく過程が示されているか。

客が一度決めたプランから気が変わり、別の選択肢を選ぶ場面があるか。

客が最終的に選んだプランの詳細について店員と相談する場面があるか。

項目を追加 | 空欄の項目を削除 | 自動チェックを実行 | チェック結果を全て削除

客が一度決めたプランから気が変わり、別の選択肢を選ぶ場面があるか。

[詳細を開く] 評価: True

客が最終的に選んだプランの詳細について店員と相談する場面があるか。

[詳細を開く] 評価: False

根拠:

- 客 (assistant) は最終的に箱根の温泉と観光のプランに興味を示していますが、具体的な日程や予約の詳細についてはまだ決まっていません。
 - 「箱根の温泉と観光、魅力的ですね。宿泊施設の詳細を教えてください。」
 - 「露天風呂のある旅館が良いですね。料金や空き状況を教えてください。」
 - 「具体的な日程はまだ決まっていますが、冬の週末を考えています。おすすめの日はありますか？」
 - 「1月中旬か2月初めが良さそうです。具体的な日程を考えてみます。」
- 店員 (auto debugger) は、具体的な日程が決まったら連絡するように促していますが、最終的なプランの詳細についての相談は行われていません。
 - 「それでは、1月中旬や2月初めで具体的な日程が決まったら、ぜひ教えてください。空き状況を確認して、予約をお手伝いしますね。」

以上のやり取りから、客が最終的に選んだプランの詳細について店員と相談する場面はありません。

図 A-8 自動評価コンポーネント。上部のように評価項目が確認・編集できる。対話結果がある状態で LLM によるチェックを実行すると、下部のように判定結果が追加される。