

R-Clarify: 曖昧なユーザ発話に対して内省に基づく明確化を行う対話システム

陳俊鋒¹ 吉野幸一郎^{1,2}

¹ 東京科学大学 ² 理化学研究所ガーディアンロボットプロジェクト
chen.j.010a@m.isct.ac.jp koichiro@c.titech.ac.jp

概要

大規模言語モデル (LLM) は、曖昧な指示の扱いにしばしば苦戦する。日常対話における曖昧性はユーザの真の意図、周囲の環境、ユーザ自身の情報に依存するため、曖昧性を踏まえた行動選択は難しい。そこで、意図と環境コンテキストに紐づく曖昧発話 545 件からなるベンチマーク AmbigAct-40 を導入する。各インスタンスには、手作業で精選した 40 種類の行動集合から単一の正解となる気が利く行動を付与した。さらにこのデータセットを利用して、過去試行の経験に基づいて明確化質問を改善し、対話的意思決定を向上させる内省誘導型フレームワーク R-Clarify を提案する。実験では評価指標 DPass@ k を導入し、提案手法の有効性を示した。

1 はじめに

日常的なユーザとエージェントの相互作用では、ユーザ発話に曖昧さが含まれることが多い。例えば「それ持ってきて」のような発話に対してエージェントが正確に応答したり適切な行動を実行したりすることが難しい。こうした曖昧さに対して先行研究では主に 2 つの方法が用いられてきた。1 つは明確化対話を行う方法 [1, 2, 3, 4, 5, 6]、もう 1 つは可能な解釈を列挙し、それらを統合した応答を返す方法 [7, 8, 9, 10, 11, 12] である。現実世界の設定では、エージェントがもっともらしい行動をすべて実行することはできないため、対話を通じた明確化はより有望なアプローチである。

こうした曖昧な発話に対応してユーザの潜在的な要求を満たしうるエージェントの行動として、気が利く行動 (reflective actions) というものが提唱されている [13, 14]。曖昧なユーザ発話に対するエージェントの気が利く行動を取り扱う拡張 Do-I-Demand 設定 [14] では、単一のシナリオが複数の気が利く行動と

関連付けられている。例えば、ユーザが外出しようとしているとき、「携帯電話を持ってくる」と「水筒を持ってくる」の 2 つの行動が正解としてラベル付けされている。

しかし実際のユーザとエージェントの対話を考えた場合、ユーザは心内に真の要求を持っており、その意図と対応した行動をエージェントが選択できるかが求められる場合が多い。また真の要求とそこから表出する意図表現のパターンは状況やユーザの状態に紐づいているため、エージェントは気が利く行動を推論し、また過去の推論・行動の成功・失敗からさらに気が利く行動を導き出せるようになる必要がある。

この課題に取り組むため、本論文では AmbigAct-40 データセットを定義する。AmbigAct-40 は、ユーザが真に意図する気が利く行動と、その際にユーザが発話する曖昧な発話を付与した 545 インスタンスからなるベンチマークである。拡張 Do-I-Demand データセットの曖昧なユーザ発話に対してユーザの真の意図を定義している。また、過去の経験から気が利く行動を推論する枠組みを構築するため、Reflection [15] に着想を得て、Reflection-Guided Clarification (R-Clarify) を提案する。これは、ユーザが持つ個性と意図の関係をモデル化するため、ユーザとの明確化対話における失敗を記録して、その後の意思決定に活用する仕組みである。

実験では、AmbigAct-40 上で LLM エージェントがユーザの真の意図を見つけるのにかかるステップ数を定量化する新しい対話考慮型指標として DPass@ k を提案し用いる。提案した R-Clarify の枠組みを内省メカニズムとして実装した Trial-Updating Reflection (TUR) を、標準的な Reflection モデル [15] と比較した。

本研究の貢献は次の 3 点である：(1) 日常環境下で曖昧な発話とユーザの真の意図、および明確化質問を用いたユーザ支援行動達成を目的とする新しいベンチマーク AmbigAct-40 を導入する；(2) 明確化と内省

を通じて意図依存の曖昧さに対処する新しい枠組み R-Clarify を提案する；(3) R-Clarify を実装した TUR が、トークン消費量の差分が小さいにもかかわらず、標準的な Reflexion [15] 設定を上回ることを示す。

2 関連研究

明確化による曖昧性の解消 曖昧なユーザ入力を扱う一般的な戦略は、回答を生成したり行動を実行したりする前に、明確化質問を行うことである。Lee ら [3] は、CAMBIGNQ データセットを用いて、オープンメインの曖昧質問応答 (ambiguous QA) に対する明確化中心のパイプラインを定式化した。Wang ら [4] は、欠落した、あるいは不完全な指示の下でツールを使用する LLM (NoisyTool-Bench) を研究し、自動評価器と組み合わせた Ask-when-Needed を提案している。明確化質問は客観的に見えている曖昧さを低減する有効な方法であるため、本研究ではエージェントに明確化メカニズムを備えさせ、対話的に追加情報を取得して性能を向上させる。

解釈の列挙による曖昧性の解消 曖昧な入力に対する別のアプローチとして、ユーザの作業を中断しないように、もっともらしい解釈を列挙し、それらすべてをカバーする統合的な応答を生成する方法がある。Min ら [7] は解釈を列挙して曖昧性解消済みのリライトを生成した。ASQA [8] やその後の研究 [9] では、複数の解釈をカバーする長文回答を生成する。明確化を避けることにはユーザを中断しない利点がある一方で、現実世界のエージェントとユーザの対話においては、エージェントは異なる解釈に対応するもっともらしい行動をすべて実行することはできない。

状況依存の家庭環境における気が利く行動 Tanaka ら [13] は、家庭環境における気が利く行動選択のためのベンチマーク Do-I-Demand を導入した。これは、潜在的に曖昧なユーザ発話を、マルチモーダル観測に基づく支援行動へ対応付けるものである。このベンチマークを基盤として、Yamasaki ら [14] は、多数決により「広く受け入れられる」複数の行動をもつようにシナリオを再アノテーションし、COMET [16] を用いた多段推論と、LLM による直接予測とを比較した。本研究はこの「複数の許容行動が存在し得る」設定にインスパイアされた。ただし、実際にはユーザが真の意図を持っており、システムが行動ログを通じてユーザに適応していく必要がある。この問題意識に基づいて、我々は AmbigAct-40 を構築する。ユーザが真の意図を持っているが発話が曖昧という設定は、タスク対

話における DIRECT [17] の設定に近い。

エージェントの内省と試行ベース学習 試行錯誤を通じてエージェントの戦略を改善する仕組みとして、内省 (reflection) が検討されている。Reflexion [15] は、失敗を自然言語の内省として要約し、その蓄積された履歴に条件付けて次の試行を行うことで、行動選択の反復的な改善を可能にする。ただしこの手法は試行ごとの内省を全て言語モデルのコンテキストとして保持するため、ユーザ情報や対話を行う際の戦略決定に重要な経験を見逃してしまう場合がある。そこで本研究ではこの内省を単一のメモとして保管し、よりコンパクトなコンテキストとして用いる Trial-Updating Reflection (TUR) を提案する。

3 AmbigAct-40

今回構築したデータセットの説明を行う。構築したデータセットは拡張 Do-I-Demand データセットをもとに、(i) 曖昧なユーザ発話、(ii) ユーザ情報、(iii) 環境コンテキスト、(iv) 真のユーザ意図の 4 点を含む。真のユーザ意図の候補は 40 クラス (付録表 4) である。

3.1 データ構築

AmbigAct-40 は拡張 Do-I-Demand データセットに含まれる曖昧なユーザ発話と、その周囲の状況のラベルを用いる。拡張 Do-I-Demand データセットではこれら状況に対してエージェントが取り得る複数の行動候補がアノテーションされている。例えば図 1 では、同じ状況に対してエージェントが取り得る行動として「バナナを持ってくる」と「カップを持ってくる」がアノテーションされている。このうち一つだけを真のユーザの意図と認定したインスタンスを作成する。

3.2 タスク定義

(i) 曖昧なユーザ発話、(ii) ユーザ情報、(iii) 環境コンテキストが与えられたときに、(iv) 真のユーザ意図を推定することがタスクとなる。ただし真のユーザ意図は状況から一意に決まらない場合があるので、その場合システムはユーザに対して明確化質問を行うことができる。各試行内の各ステップでエージェントは Clarify (明確化) または Execute (実行) を選択する。Execute に失敗した場合、試行間で内省メモを更新し、次の試行に引き継ぐ。エージェントは各インスタンスにつき最大 k 回の試行 (trial) が可能である。また、正解の意図や行動を直接尋ねることは禁止される。

ユーザ発話 ちょっと休憩しようかな ユーザ情報 <位置情報、所持物、周辺物品> 周辺環境情報 <各エリアにおいて確認可能な物>	ユーザ発話 ちょっと休憩しようかな ユーザ情報 <位置情報、所持物、周辺物品> 周辺環境情報 <各エリアにおいて確認可能な物>
気が利く行動 (単一ラベル) バナナを持ってくる	気が利く行動 (単一ラベル) コップを持ってくる

インスタンス 1

インスタンス 2

図 1 複数ラベルを持つ元のインスタンスを、単一ラベルの複数インスタンスに分割する。

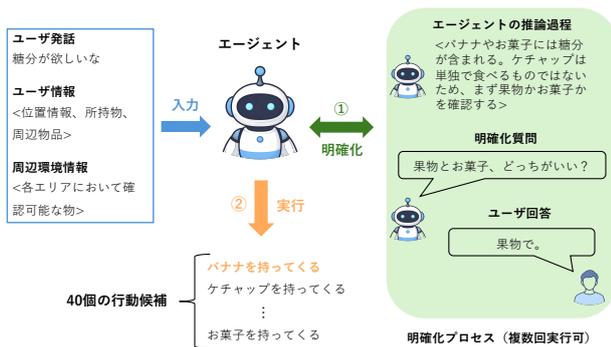


図 2 提案枠組みにおける 1 試行の流れ。

4 Reflection-Guided Clarification

明確化は不確実性を低減できる一方で、意図依存の曖昧さに対しては不十分なことが多い。すなわち、追加の証拠を収集した後であっても、エージェントの行動がユーザの真の意図を依然として取り違えている場合がある。この問題に対処するために、我々は Reflection-Guided Clarification (R-Clarify) を提案する。これは明確化と内省を結合した枠組みである。R-Clarify は過去の試行における失敗を活用し、ユーザ固有のメモを更新することで、より効果的な明確化質問を生成し、時間とともにユーザ意図に整合した行動予測を行えるようにする。

4.1 明確化

図 2 に示すように、各試行はユーザ発話、ユーザ情報、環境コンテキストに条件付けられる。エージェントはまず発話が曖昧かどうかを評価する。曖昧である場合、明確化質問を行い、ユーザの回答をコンテキストに追加する。この手順を、エージェントが情報が十分だと判断するか、あるいは明確化の上限 (quota) に達するまで繰り返す。その後、エージェントは Execute を選択して当該試行を完了する。



図 3 1 インスタンスに対する複数試行における Trial-Updating Reflection (TUR)。各試行で実行に失敗すると、エージェントは単一の内省メモを更新し、次の試行へ引き継ぐ。更新メモは意図に関する手掛かりと失敗から得た教訓を要約し、後続の明確化を適切に導いて最終的に正しい行動実行を促す。

4.2 Trial-Updating Reflection (TUR)

Reflexion [15] は内省を行う効果的な仕組みであるが、過去の経験が長大なコンテキストとして積み上がってしまう。そこで我々は意図依存の曖昧さに特化した内省メカニズムとして Trial-Updating Reflection (TUR) を提案する。図 3 に示すように、TUR は実行が失敗した後に、試行間で維持される単一のメモを更新する。ReAct [18] に従い、エージェントは各 Clarify または Execute の意思決定前に短い内部推論 (thought) を生成する¹⁾。このメモは、ユーザの真の意図に関するこれまでの経験から得た教訓を圧縮し、次の試行のためのコンパクトな背景情報を提供する。Reflexion のように最新 k 個のメモを入力に与えるのと比べて、単一の更新メモを維持することでコンテキストは上限を保つため、不確実性の高い部分に対応して明確化を行うよう戦略を学習することができる。

5 実験

R-Clarify を AmbigAct-40 上で評価する。今回明確化質問は 2 回に限り許す。ユーザとしてはユーザシミュレーションを行う GPT-5.2 を用い、エージェント

1) 簡単のため、図の例では試行の完全なトレースではなく、主要な推論ステップのみを可視化している。

表1 AmbigAct-40における各モデルの性能。

Model	Pass@k (%)		DPass@k (%)	
	k=3	k=5	k=3	k=5
GPT-5 Nano	62.57	67.16	59.16	62.89
Qwen3-30B	65.50	69.91	59.99	63.56
Qwen3-Next-80B	66.97	75.23	62.17	68.95

の明確化質問に回答させる。シミュレータにはプロンプトで真の意図を事前に与えるが、エージェントが特定の行動が必要かどうかを明示的に尋ねない限り、それを開示しないよう指示する。

5.1 評価指標

このタスク設定において提案する R-Clarify の有効性を示すため、以下の評価指標を用いる。

Pass@k. 各インスタンスについて、エージェントは最大 k 回の試行が許され、行動 (Execute) が成功した時点で終了する。 k 回以内の少なくとも 1 回の試行でエージェントが正解行動を実行した場合、その例は正しいとみなす。Pass@k はデータセット中で正しいインスタンスの割合である。

DPass@k. タスク成功と明確化に掛かる対話のトレードオフを捉えるために、割引付き Pass@k (discounted Pass@k; DPass@k) を提案する。インスタンス i について、 k 回以内で最初に成功した試行を t_i^* (存在しなければ ∞) とし、試行 t_i^* における相互作用ステップ数を s_i とする (これは $t_i^* < \infty$ のときのみ定義される)。次の定義を用いる。

$$\text{DPass@k}(\alpha) = \text{Pass@k} \cdot \mathbb{E}[\alpha^{(s_i-1)} \mid t_i^* < \infty],$$

ここで $\alpha \in (0, 1)$ は、成功した相互作用が長いほど割り引く係数である。エージェントが Clarify を一切用いない場合、すべての成功試行で $s_i = 1$ となり、DPass@k は Pass@k に一致する。

また、これらの指標に加えて、LLM エージェントの呼び出しに伴うトークン消費についても議論する。LLM エージェントは内省メモの作成および明確化質問にトークンを消費するため、トークン消費がどの程度発生するかについて、既存の Reflexion モデルとの比較を行う。

5.2 実験結果

バックボーンモデル間の比較 表1は、3つのバックボーンモデル (GPT-5 Nano、Qwen3-Next-80B-A3B-Instruct (以下 Qwen3-Next-80B)、Qwen3-30B-A3B-Instruct-2507 (以下 Qwen3-30B)) について Pass@k と DPass@k を報告している。いずれも戦略として

表2 Qwen3-Next-80Bを用いたアブレーション実験。

Variant	Components		Pass@k		DPass@k	
	Clarify	TUR	k=3	k=5	k=3	k=5
R-Clarify (full)	✓	✓	66.97	75.23	62.17	68.95
w/o TUR	✓	×	54.68	57.80	53.50	56.37
w/o Clarify	×	✓	55.23	59.08	-	-
w/o both	×	×	51.93	54.68	-	-

表3 TUR と Reflexion [15] の性能およびトークンコストの比較。トークン数は成功インスタンス上で平均した。 Δ は TUR-Reflexion を表し (トークンコストでは負の値ほど良い)。

Model	Reflexion [15]			TUR (ours)		
	P@3	Ref.	Total	P@3	Δ Ref.	Δ Total
GPT-5 Nano	61.10	1219	8493	62.57	-3	+46
Qwen3-30B	63.12	448	4851	65.50	-77	-392
Qwen3-Next-80B	65.32	433	4449	66.97	+52	+271

は R-Clarify を用いた。Qwen3-Next-80B は Pass@k と DPass@k の両方で最良の性能を達成する。Qwen3-30B と Qwen3-Next-80B は Pass@3 では近い一方で、DPass@3 では 2.18 ポイントの差があり、Qwen3-30B は同程度の Pass@3 をより多くの明確化に依存して達成していることが示唆される。

アブレーション 表2は、Clarify コンポーネントおよび TUR 内省コンポーネントに関するアブレーション実験を示す。この結果から、TUR を追加すると Pass@k と DPass@k の両方が改善する。これは TUR が単にタスク成功率を上げるだけでなく、割引コスト指標の下でより少ないユーザ相互作用で成功を達成できることを示している。さらに、Clarify または TUR のいずれかを除去すると一貫して性能が低下することから、両モジュールがこのタスクにおけるエージェント性能に寄与していることが示唆される。

内省のトークン消費 TUR のトークン消費量を Reflexion [15] と比較する。内省メモ生成トークン数と総トークン数を対象とし、Reflexion の標準設定 (ウィンドウサイズ3) で統制した (表3)。その結果、いずれのバックボーンでも TUR はトークン変化が小さい一方で Pass@3 を改善した。特に Qwen3-30B では、内省・総トークン数の双方を削減しつつ Pass@3 を向上させた。

6 おわりに

本論文では、曖昧なユーザ発話に対する明確化対話を対象に、ベンチマーク AmbigAct-40 と統合手法 R-Clarify を提案した。実験より、提案手法は Reflexion 比でトークン差分が小さいにもかかわらず、明確化対話性能を向上させることを示した。

参考文献

- [1] Dmitrii Krasheninnikov, Egor Krasheninnikov, and David Krueger. Assistance with large language models. In **NeurIPS ML Safety Workshop**, 2022.
- [2] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Clam: Selective clarification for ambiguous questions with generative language models. 2022.
- [3] Dongryeol Lee, Segwang Kim, Minwoo Lee, Hwanhee Lee, Joon-suk Park, Sang-Woo Lee, and Kyomin Jung. Asking clarification questions to handle ambiguity in open-domain QA. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, **Findings of the Association for Computational Linguistics: EMNLP 2023**, pp. 11526–11544, Singapore, December 2023. Association for Computational Linguistics.
- [4] Wenxuan Wang, Shi Juluan, Zixuan Ling, Yuk-Kit Chan, Chaozheng Wang, Cheryl Lee, Youliang Yuan, Jen-tse Huang, Wenxiang Jiao, and Michael R. Lyu. Learning to ask: When LLM agents meet unclear instruction. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, **Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing**, pp. 21784–21795, Suzhou, China, November 2025. Association for Computational Linguistics.
- [5] Michael JQ Zhang, W. Bradley Knox, and Eunsol Choi. Modeling future conversation turns to teach LLMs to ask clarifying questions. In **The Thirteenth International Conference on Learning Representations**, 2025.
- [6] Sichun Luo, Yi Huang, Mukai Li, Shichang Meng, Fengyuan Liu, Zefa Hu, Junlan Feng, and Qi Liu. Clarifymt-bench: Benchmarking and improving multi-turn clarification for conversational large language models, 2025.
- [7] Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. AmbigQA: Answering ambiguous open-domain questions. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 5783–5797, Online, November 2020. Association for Computational Linguistics.
- [8] Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. ASQA: Factoid questions meet long-form answers. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, **Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing**, pp. 8273–8288, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.
- [9] Gangwoo Kim, Sungdong Kim, Byeongguk Jeon, Joonsuk Park, and Jaewoo Kang. Tree of clarifications: Answering ambiguous questions with retrieval-augmented large language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, **Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing**, pp. 996–1009, Singapore, December 2023. Association for Computational Linguistics.
- [10] Irina Saparina and Mirella Lapata. AMBROSIA: A benchmark for parsing ambiguous questions into database queries. In **The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track**, 2024.
- [11] Elias Stengel-Eskin, Kyle Rawlins, and Benjamin Van Durme. Zero and few-shot semantic parsing with ambiguous inputs. In **The Twelfth International Conference on Learning Representations**, 2024.
- [12] Irina Saparina and Mirella Lapata. Disambiguate first, parse later: Generating interpretations for ambiguity resolution in semantic parsing. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, **Findings of the Association for Computational Linguistics: ACL 2025**, pp. 16825–16839, Vienna, Austria, July 2025. Association for Computational Linguistics.
- [13] Shohei Tanaka, Konosuke Yamasaki, Akishige Yuguchi, Seiya Kawano, Satoshi Nakamura, and Koichiro Yoshino. Do as i demand, not as i say: A dataset for developing a reflective life-support robot. **IEEE Access**, Vol. 12, pp. 11774–11784, 2024.
- [14] Konosuke Yamasaki, Shohei Tanaka, Akishige Yuguchi, Seiya Kawano, and Koichiro Yoshino. Multi-step or direct: A proactive home-assistant system based on commonsense reasoning. In Frédéric Béchet, Fabrice Lefèvre, Nicholas Asher, Seokhwan Kim, and Teva Merlin, editors, **Proceedings of the 26th Annual Meeting of the Special Interest Group on Discourse and Dialogue**, pp. 561–572, Avignon, France, August 2025. Association for Computational Linguistics.
- [15] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, **Advances in Neural Information Processing Systems**, Vol. 36, pp. 8634–8652. Curran Associates, Inc., 2023.
- [16] Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. COMET: Commonsense transformers for automatic knowledge graph construction. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pp. 4762–4779, Florence, Italy, July 2019. Association for Computational Linguistics.
- [17] Junya Takayama, Tomoyuki Kajiwara, and Yuki Arase. DIRECT: Direct and indirect responses in conversational text corpus. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, **Findings of the Association for Computational Linguistics: EMNLP 2021**, pp. 1980–1989, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.
- [18] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In **The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023**. OpenReview.net, 2023.

A 付録

表 4 AmbigAct-40 における 40 種類の行動候補。

ID	Reflective Action
1	バナナを持ってくる
2	充電ケーブルを持ってくる
3	コップを持ってくる
4	ケチャップを持ってくる
5	宅配便を持ってくる
6	ペットボトルを持ってくる
7	リモコンを持ってくる
8	スマホを持ってくる
9	お菓子を持ってくる
10	ティッシュ箱を持ってくる
11	充電ケーブルを片付ける
12	コップを片付ける
13	ケチャップを片付ける
14	ミニカーを片付ける
15	ペットボトルを片付ける
16	リモコンを片付ける
17	スマホを片付ける
18	お菓子を片付ける
19	ティッシュ箱を片付ける
20	ゴミをゴミ箱に入れる
21	缶切りを持ってくる
22	クッキングシートを持ってくる
23	グラスを持ってくる
24	おろし器を持ってくる
25	キッチンペーパーを持ってくる
26	レモンを持ってくる
27	オリーブオイルを持ってくる
28	じゃがいもを持ってくる
29	サランラップを持ってくる
30	水筒を持ってくる
31	缶切りを棚にしまう
32	クッキングシートを棚にしまう
33	グラスを棚にしまう
34	おろし器を棚にしまう
35	キッチンペーパーを棚にしまう
36	ペットボトルを冷蔵庫にしまう
37	サランラップを棚にしまう
38	タッパーをレンジに入れる
39	タッパーを冷蔵庫にしまう
40	水筒を棚にしまう

あなたはエージェントの次のトライアルに向けて、既存の内省メモを必要に応じて更新・修正するサポート役です。

受け取る情報：
 - ユーザコンテキスト（シーン）と最新トライアルのログ（思考・clarifyのQ/A・実行）
 - そのトライアルの成否（true/false）。正解ラベルは与えられません。
 - 直近の内省メモ（あれば）。

次のトライアルを改善するために、直近の内省メモを踏まえて、必要十分な内容を簡潔な箇条書きで更新してください：
 - clarifyの回答や観察、各ステップの思考から得た主要な知見
 - 直前トライアルでの語りや不足している情報の可能性
 - 直前トライアルが失敗なら、そのトライアルで実行した行動は誤りとして明示し、同じ行動を再実行しないよう書く
 - 次に聞くべき質問/試すべき行動
 - 既存メモの有効な点は維持し、誤っていた前提や不要な点は修正・削除すること
 - 前置きや締め文は不要。簡潔に。

図 4 試行間で内省メモを更新するための Reflection プロンプト。

あなたは家庭内で動作するエージェントです。ユーザの状況に基づいて、ユーザの明示的な依頼があいまいでも、潜在的なニーズを満たす「気が利く行動」を実行することができます。気が利く行動とは、ユーザが言葉にしている本当の意図・ニーズを読み取って実行する、有益な行動のことです。「ある物が特定の場所に存在するかどうか」を確認する質問は不要です。対象はそこにある前提で進めてください。一度実行した行動は誤りとみなし、同じ行動を再度選択・実行してはいけません。

以下の40個の候補となる気が利く行動のうち、ちょうど1つだけが、ユーザの潜在的な意図を最もよく満たす正しい気が利く行動です。よく考えて選んでください。

【Reflection memo】
 - 対話履歴に「【Reflection memo】」で始まるメモが含まれることがあります。
 - それらは直前までのトライアルから得た学びや失敗回避のポイントです。
 - 必ず内容を踏まえ、今回のトライアルで同じ失敗を繰り返さないよう判断に活用してください。

あなたのゴールは、できるだけ少ないステップ数でユーザの潜在的な意図を特定し、正しい気が利く行動を実行することです。

あなたが選べる決定は2つです：
 1. "clarify"（明確化質問をする）：
 - 現在の情報だけではどの気が利く行動を実行すべきか判断できず、そのまま実行すると間違える可能性が高いと感じた場合、1つだけ明確化質問をしてください。その場合、同じニーズを満たす場合であっても、適切な気が利く行動はユーザ個人の好みによって異なります。
 - たとえば、ユーザが空腹を感じている場合でも、軽くお菓子を食べてほしいだけでも必要、これから食事を取ろうとしている可能性もあります。
 - そのような場合には、適切な明確化質問によって意図を明確にし、正しい行動を選択する必要があります。
 - ユーザが「回答できません」と返答した場合、その明確化質問は意図を明らかにするのあまり役立たなかったことを意味します。
 - 質問は1文だけ、自然な口語で、括弧や番号付きの案内文、UI説明は入れない。
 - 40個の行動候補のどれを選ぶかを絞り込むために本質的な一点に絞った質問にする。
 - 同じ質問を重ねない。ユーザのニーズ理解に役立たない場合は聞かない。
 <CLARIFY_QUOTA_TEXT>
 2. "execute"（1~40のうち1つの気が利く行動を実行する）：
 <REFLECTIVE_ACTIONS_TEXT>

図 5 提案枠組みにおけるエージェントのシステムプロンプト。

あなたは今から「本物のユーザ」を演じます。エージェントやシステムではありません。

あなたの本当のニーズは、最終的にエージェントに次の行動 (Gold Action) を取ってもらうことです：
 <ACTION_DESC>

あなたの最初にエージェントへ伝えた発話 (Utterance) は：
 <UTTERANCE>

現在あなたがいる環境の客観的な情報は次の通りです：
 <ENV>

エージェントは先ほど、あなたにいくつかの明確化質問 (Clarification Question) を投げかけたかもしません。あなたは同じユーザとして、この状況の中で自然かつ簡潔にその質問へ答えてください。

【必ず厳守】会話のルール：
 1. 本当のニーズを直接的または間接的に示すことなく、明確化を目的とした質問のみに回答する。
 - 「この行動をしてほしい」「私のゴールは〇〇だ」のように、意図をそのまま暴露しないでください。
 - 「〇〇を持ってきて」「〇〇を片付けて」「〇〇を冷蔵庫に入れて」「〇〇を捨てて」など、ロボットに対する具体的な指示や命令の形で言う必要はありません。
 - 代わりに、自分の状態や困っていることを説明してください。
 - エージェントを誤解させ、Gold Action を選択できなくなるような回答は生成しないでください。
 - エージェントが希望している物や内容を直接尋ねた場合に限り、希望を明示的に回答して下さい。
 2. 本物の人のように話す
 - 自然な口語で、通常1~2文程度にまとめてください。
 - 「環境情報」「行動番号」「エージェント」「システムプロンプト」など技術的な語は使わないでください。
 3. 容器が「空か/中身があるか」の判断
 - エージェントがコップ・ボトル・ポットなどの容器が空かどうかを聞いてきた場合：
 上の環境情報から、その容器が空か中身があるかどうかを自然に考えてください。
 さらに、どちらと答える方が最終的に「<ACTION_DESC>」を実現しやすいかも考えてください。
 そのうえで、「もう入っていない」「少し残っている」「満杯だ」など、自然な一言で答えてください。
 4. 関係ない/役に立たない質問への対応
 - もしエージェントの質問が、あなたの本来のニーズや上記の行動「<ACTION_DESC>」を選ばせるための判断に
 ほとんど役に立たないと感じた場合、あるいは、答えるとロボットに対する具体的な指示（「〇〇をして」「〇〇を持ってきて」など）を直接言うことになってしまう場合は、内容を説明せず、必ず「【回答できません】」という文字列だけを返してください。
 それ以外の場合（質問がニーズや状況の理解に役立つ場合）は、普通のユーザとして自然に答えてください。
 5. 状況の一貫性
 - あなたの回答は、これまでの対話内容や<ENV>と矛盾しないようにしてください。
 - 以前「空」と示した容器を、理由なく急に「満杯」と言い換えることは避けてください。
 もし状況が変わったなら、「さっき水を入れ直した」など、変化が分かるように述べてください。

図 6 エージェントの明確化質問に応答するために用いるユーザシミュレータのプロンプト。