Learning to Reason and Use Tools through Unsupervised Fine-Tuning in Task-Oriented Dialog Systems

Aprendiendo a razonar y usar herramientas mediante fine-tuning no supervisado en sistemas de diálogo orientados a tareas

Markel Ferro,[1] **Oier Lopez de Lacalle**[1]
[1]HiTZ Center - Ixa, University of the Basque Country UPV/EHU
{markel.ferro, oier.lopezdelacalle}@ehu.eus

Abstract: Current dialogue systems struggle with dynamic information retrieval, often leading to hallucinations and lower response accuracy. We address this by adapting the ReAct framework for Task-Oriented Dialogue, enabling Large Language Models (LLMs) to access external knowledge and produce factual responses. Mainly, we propose an unsupervised fine-tuning pipeline that harvests reasoning trajectories via in-context learning inference. High-quality samples are filtered using an LLM-based judge to construct a robust training set. This is enhanced by a unsupervised self-improvement loop, where improved checkpoints generate increasingly better trajectories for subsequent fine-tuning iterations. Experiments on the SIMMC dataset demonstrate that ReAct-based systems outperform baselines due to superior reasoning and tool use. Notably, our fine-tuned 8B model surpasses a 70B in-context system. Finally, we present an error analysis, impact of scene complexity, and cross-domain generalization.
Keywords: Dialog Systems, Natural Language Generation, Unsupervised Learning.

Resumen: Los sistemas de diálogo actuales tienen problemas con la recuperación dinámica de información, causando alucinaciones y menor precisión. Abordamos este problema adaptando ReAct al Diálogo Orientado a Tareas, permitiendo a los Grandes Modelos de Lenguaje (LLM) usar conocimiento externo y generar respuestas fácticas. Proponemos un proceso de fine-tuning no supervisado que recopila trayectorias de razonamiento mediante inferencia de modelos adaptados en contexto. Un juez LLM filtra las muestras de alta calidad para construir un conjunto de entrenamiento robusto. Esto se potencia con un bucle de automejora no supervisado, donde los checkpoints refinados generan trayectorias superiores para iteraciones posteriores. Experimentos en SIMMC demuestran que los sistemas ReAct superan a los modelos de referencia gracias a un mejor razonamiento y uso de herramientas. Destacamos que nuestro modelo de 8B ajustado supera a un sistema en contexto de 70B. Finalmente, analizamos los errores, la complejidad de la escena y la generalización multidominio.
Palabras clave: Sistemas de Diálogo, Generación de Lenguaje Natural, Aprendizaje No Supervisado.

1 Introduction

Task-oriented Dialogue (TOD) systems aim to accomplish predefined tasks, such as booking a flight or retrieving product information, through natural language interactions with users (Wang et al., 2025). Traditionally, these systems followed a modular architecture consisting of separate components for Natural Language Understanding (NLU), Dialogue State Tracking (DST), and Natural Language Generation (NLG), among others (Celikyilmaz, Deng, and Hakkani-Tür,

2018). While effective, modular approaches are inherently complex, susceptible to error propagation between components, and require independent training of each module. Consequently, recent advances in Large Language Models (LLMs) have driven a shift toward end-to-end TOD systems, where generic LLMs are adapted to construct task-specific conversational agents (Yi et al., 2025). Although such models produce fluent and natural responses, they frequently exhibit factual inconsistencies, commonly referred to as hallucinations (Dhuliawala et al., 2023), as they rely solely on the knowledge obtained during training, and lack mechanisms for dynamically accessing real-time information necessary for accurate task completion.

To overcome these limitations, recent research has demonstrated that integrating Chain-of-Thought (CoT) reasoning (Wei et al., 2023) with tool-use in LLMs leads to strong performance in tasks requiring access to external information (Yao et al., 2023). In particular, the ReAct framework employs carefully designed prompting strategies to guide the model through an iterative sequence of *thoughts*, *actions*, and *observations*. Here, thoughts represent the model's internal reasoning process, decomposing a task into subproblems; actions denote external API calls or program executions aimed at solving those subproblems; and observations capture the responses returned by the external tools.

However, adapting the ReAct framework to TOD systems is non-trivial (Elizabeth et al., 2025), as it requires the design of appropriate tools (i.e., actions) and carefully crafted prompting strategies that enable the model to reason effectively, decomposing tasks into subproblems and selecting the correct tools for each. In this paper, we show that such an adaptation is not only feasible but can be improved through a fully unsupervised fine-tuning methodology that enhances the agent's reasoning and tool-use capabilities without human supervision. Our proposed pipeline first derives reasoning trajectories from the training data via In-Context Learning (ICL) inference. The generated trajectories are then filtered using an LLM-based judge, which selects only high-quality instances to construct a dataset for fine-tuning. This approach addresses the starting problem that agents have by autonomously generating and curating its own dataset.

Our evaluation is conducted manually on the SIMMC dataset (Moon et al., 2020; Kottur et al., 2021; Kottur and Moon, 2023), which comprises dialogues centered around objects situated in store environments. We analyze its fashion and furniture domains separately to assess how scene complexity affects system performance. To further investigate how difficulty affects performance, we introduce synthetic, *fake*, objects to simulate scenarios with increased complexity. In addition, we examine how inefficient reasoning trajectories impact the fine-tuning process in terms of both performance and computational efficiency. Finally, we evaluate the generalization ability of our system by comparing it against alternative approaches on the same dataset tasks. As a result of our experimentation, we find that:

1) Reasoning and tool use enhance information access in TOD systems. We demonstrate that incorporating explicit reasoning and tool-use mechanisms enables more effective utilization of external knowledge resources compared to approaches where information access is limited to the input context.

2) Unsupervised iterative self-improvement effectively adapts smaller models. Our proposed fine-tuning pipeline enables 8B-parameter models to match or even surpass the performance of their 70B-parameter counterparts. This result is particularly significant, as it provides the deployment advantages of smaller models (reduced computational cost and resource requirements) without compromising performance.

3) The proposed approach exhibits robust performance in complex scenarios. In settings with high scene complexity, characterized by a large number of objects, prompting-based methods experience a more rapid degradation in performance compared to our ReAct-based approach.

4) Fine-tuning on reasoning trajectories yields strong cross-domain generalization. We demonstrate that previous fine-tuning approaches often achieve in-domain performance through overfitting, resulting in poor transfer to unseen domains. In contrast, our methodology promotes better generalization across domains, effectively mitigating overfitting.

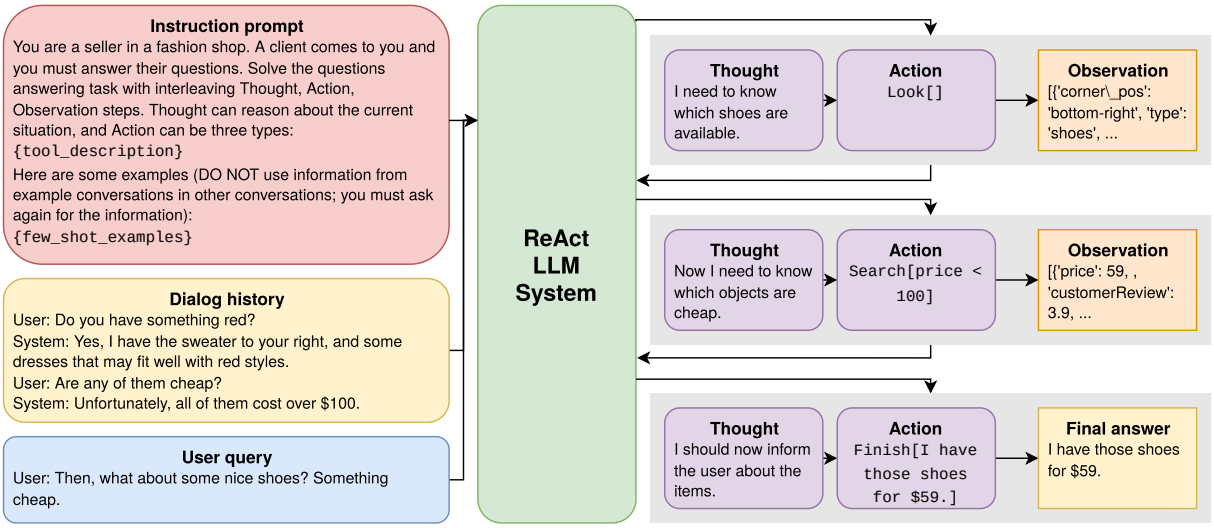

Figure 1: A diagram illustrating the ReAct LLM system as it processes a task. The system takes an Instruction prompt, Dialog history, and a User query as inputs. It then generates the Final Answer through an iterative sequence of Thoughts, Actions, and Observations.

2 *Related Work*

TOD systems with agents The rapid advancement of Large Language Models has been transformative. However, LLMs operate on static, parametric knowledge acquired during training, which results in them generating factually inconsistent or outdated information, a phenomenon known as "hallucination". This is particularly problematic in TOD systems, where providing correct information is necessary for task completion. To mitigate these shortcomings, researchers have developed agentic frameworks, such as the ReAct framework (Yao et al., 2023), that provide LLMs access to external information sources. Recent research has focused on developing agentic TOD systems capable of reasoning, planning, and using tools (Xu et al., 2024), with some works expanding on this idea by making use of multi-agent architectures (Gupta et al., 2024) for managing the complexity of real-world dialogue. However, despite the clear potential of agentic frameworks like ReAct, their direct application to the structured and constrained nature of TOD is not trivial (Elizabeth et al., 2025), as they have a tendency for the model to simply imitate the few-shot examples in its prompt, and to generate inconsistent reasoning.

Evaluation of TOD systems The evaluation of LLM dialogue systems has long been a notoriously difficult problem. Many traditional automatic metrics were inherited from other Natural Language Processing tasks like machine translation. Metrics such as BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004), which measure the overlap between a generated response and a reference, are inadequate for assessing the quality of a conversation (Liu et al., 2016). Even metrics designed for TOD, such as Inform Rate and Success Rate, have demonstrated significant limitations with LLMs. These metrics focus on the final outcome, e.g., whether the system provided the correct entity. The evaluations mainly rely on string-matching, which can lead to incorrect evaluations if the model contains the correct attribute somewhere else in the response, even if syntactically it does not refer to the correct object (Acikgoz et al., 2025). This is further exacerbated in multi-object responses. A methodology guided using these metrics may lead to systems that appear successful on paper but unreliable in real-world scenarios.

LLM judges To address the problems of traditional metrics while avoiding the costly human evaluation, the research community has recently started using LLMs as judges. This approach aims to use the language understanding of LLMs to assess the quality of outputs generated by other models. Large proprietary models like GPT-4 are often employed for this purpose due to their high alignment with human evaluation (Gu et al.,

2025); however, their high cost and lack of controllability have spurred the development of open-source LLM judges. A state-of-the-art example is Prometheus 2 (Kim et al., 2024), an open-source model developed specifically for the task of evaluating other LLMs. It offers a high degree of controllability, supporting evaluation based on custom, user-defined criteria. It is capable of performing both direct assessment (assigning a score on a Likert scale) and pairwise ranking (determining a preference between two outputs).

Task-oriented Dialogue Datasets SIMMC (Kottur et al., 2021) and MultiWOZ (Eric et al., 2019) are prominent reference datasets in the field. They provide multi-turn conversations focused on particular tasks, ranging from simulating a police interaction to purchasing clothes. However, despite their varied nature, models trained on these datasets expose the need for dynamic reasoning capabilities (Hemanthage et al., 2023), as they are often unable to adapt to new tasks. This limitation has motivated recent interest in strategies that dynamically adapt models to novel situations.

3 ReAct as Dialogue System

Problem formulation We base our study on the SIMMC 2.1 dataset (Kottur and Moon, 2023), a task-oriented dialogue corpus designed for multimodal assistant–user interactions in realistic shopping environments. Although the benchmark defines four tasks, including coreference resolution and dialogue state tracking, this work focuses exclusively on the Automatic Response Generation task. The assistant does not have access to any ground-truth annotations, observing only the dialogue history, the current user utterance (current query) and the metadata of objects present in the scene.

The task is designed to evaluate a model's ability to generate contextually appropriate responses to user queries that require information about relevant objects within a scene. Each dialogue occurs in a specific scene containing multiple objects, and the model has access to the corresponding object metadata. This metadata includes both visual and textual representations, and the system must accurately perform metadata grounding by identifying and leveraging the objects relevant to the dialogue context.

ReAct To address this task, we design an agentic system in which the agent actively retrieves information about relevant objects. This design grounds the agent's behavior, preventing unnecessary access to unrelated information. We implement this through an adaptation of the ReAct framework (Yao et al., 2023) tailored to our dataset and equip the agent with a set of custom tools specialized for information retrieval. Figure 1 illustrates the adaptation of the ReAct framework to a dialogue system, where the LLM (depicted as the ReAct LLM system in the figure) receives the instruction prompt, dialogue history, and user query as inputs, and generates the final response through an iterative sequence of thoughts, actions, and observations triplets. Actions require the use of external tools. The tools defined in our adaptation are as follows:

- `Look[]`: It returns the visual metadata of all objects present in the scene in textual form, simulating the capabilities that a multimodal model would possess.
- `Search[query]`: A tool that retrieves the non-visual metadata of objects in the scene that satisfy the constraints specified in the query. For example, `Search[customerRating>3]` would retrieve objects with ratings greater than 3.
- `Finish[answer]`: An action used by the model to indicate the final response that the system should return. Once this action is executed, no further actions are performed. For example, `Finish["That jackets costs $300."]`.

Figure 1 illustrates an example in which the user asks for inexpensive shoes. The *thought-action-observation* triplets generated by the models (trajectory) are as follows: The system first visualizes the available shoes using the `Look[]` tool, then, it searches for cheap items in the store using `Search[]`. Finally, the system matches the data from both observations and informs the user of the relevant shoe prices using `Finish[]`.

We adapted the original prompts to our task context and further refined them based on insights from preliminary experiments. The resulting prompts are available in Appendix A.2. While the initial behavior is obtained through in-context learning, the fol-

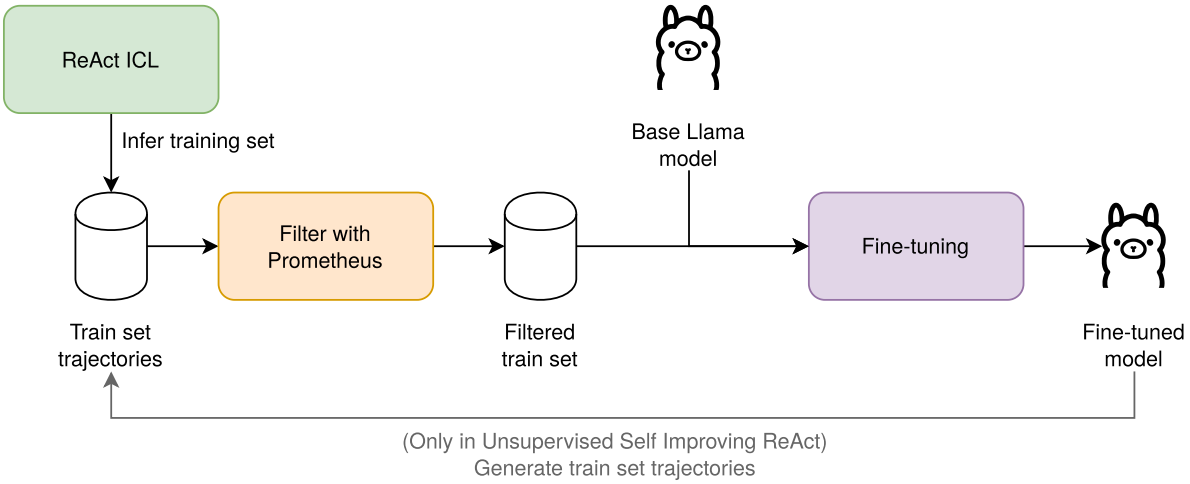

Figure 2: Diagram of the Fine-Tuning pipeline, which operates without human intervention by using model-generated instances for Unsupervised Self Improvement.

lowing section introduces a novel pipeline for unsupervised fine-tuning of the models.

4 Unsupervised Adaptation

A key challenge in fine-tuning an agent for a custom environment is the lack of reasoning trajectory data specific to that setting. Although reinforcement learning could partially address this issue, it introduces substantial additional algorithmic complexity. Consequently, the main obstacle to straightforward adaptation lies in data scarcity. To overcome this limitation, we propose a novel approach, illustrated in Figure 2. The figure shows a self-learning framework in which the model iteratively improves using its own generated reasoning trajectories. Our method consists of three main stages: generation, filtering, and fine-tuning. We first leverage the in-context learning capabilities of the system to automatically generate a training dataset containing *correct* thought–action–observation trajectories. We assume that trajectories leading to correct final responses encapsulate valid reasoning patterns. After filtering for high-quality examples, we fine-tune the model on these automatically derived trajectories.

We employ an LLM-based judge to evaluate and select high-quality trajectories for fine-tuning automatically. To automate this filtering without human intervention, we employ Prometheus 2 (Kim et al., 2024), an open-source LLM specifically aligned and trained to act as a fine-grained evaluator. It assesses the factual accuracy of the final responses, and we retain only those trajectories that receive a perfect score of 5 on its Likert scale. These threshold values were determined through preliminary experiments but can be regarded as tunable hyperparameters depending on the specific task and dataset.

Once the filtered dataset is obtained without any human supervision, we perform a supervised fine-tuning the model using LoRA (Hu et al., 2021) to accommodate resource constraints. The best-performing checkpoint is selected during fine-tuning, taking Prometheus as the evaluation metric. We use the validation split for checkpointing. The model we obtain at this stage (first iteration) is ReAct FT.

Using Prometheus to evaluate the fine-tuned model allows us to incrementally expand the set of high-quality reasoning trajectories. The key idea behind our approach is that as the model improves through fine-tuning, it can, in turn, generate better trajectories than those produced in the initial ReAct ICL 70B stage. We iteratively fine-tune the base model (Llama 8B) on the progressively enriched trajectory set until the process converges (i.e., when Prometheus detects no further performance gains). The final model obtained through this iterative process is referred to as Unsupervised Self-Improving ReAct (USI ReAct 8B). Figure 2 summarizes the complete unsupervised fine-tuning pipeline.

5 *Experimental setting*

Dataset and hyperparameters As noted in Section 3, we evaluate our models on the SIMMC 2.1 dataset, which comprises 11,244 task-oriented dialogues totaling 117,236 utterances. Our experiments focus on dialogue response generation conditioned on the dialogue history and the available scene metadata. In our experiments, the models are provided with the two preceding dialogue turns as context.

The dataset is divided into two domains: fashion and furniture. The fashion domain contains a larger number of dialogues and exhibits substantially higher scene complexity. Specifically, fashion scenes include an average of 30 objects, compared to 8.9 objects in the furniture domain. Moreover, each fashion object is described by 11 attributes,[1] whereas furniture objects have only 7.[2] This difference in both object density and metadata richness significantly increases the load and context-window saturation for the models, making the fashion domain more challenging. We define this as "scene complexity".

Each dialogue in the dataset is paired with a corresponding scene characterized by a specific object composition and associated metadata. In addition, the dataset includes information for the utterances themselves, such as the intent.[3] We transformed the visual information from the rendered environments into textual metadata, in order to enable experimentation with unimodal models.

Based on preliminary analyses, we selected Llama 3.3 70B Instruct and Llama 3.1 8B Instruct (Grattafiori et al., 2024) as the backbone models for the 70B and 8B versions of our ReAct systems, respectively. For the ICL, each model is provided with five manually curated reasoning trajectories as few-shot examples. When fine-tuning, we employ LoRA (Hu et al., 2021) with a rank of 32 to accommodate resource constraints.

Baselines To evaluate the performance of our system, we define two baseline approaches. The first, referred to as the Blind baseline, excludes all object-related information. The second, the All-in-context baseline, provides the model with the complete set of object metadata in the input. Both baselines utilize the Llama 3.3 70B Instruct model. The prompts are available in Appendix A.1.

While evaluating the proposed system against state-of-the-art models would be ideal, their training procedures render a direct comparison problematic. These models are frequently overfitted for specific datasets, a limitation that we intend to overcome with our proposal. Section 7.5 explores these generalization constraints in depth, illustrating the inability of current SOTA methods to effectively generalize outside of their training distributions.

Evaluation We opted not to use traditional automatic metrics (e.g., BLEU or BERTScore) for the final evaluation, as they fail to capture the factual nuances required for this task. Metrics designed for machine translation reward high lexical overlap; consequently, they may assign high scores to a generated response that is syntactically identical to the reference but hallucinates a crucial detail, such as stating a price of $30 instead of $300. At the same time, it could assign lower scores to fully correct answers that are shown in a different style. For the final evaluation, we adopted a blind manual annotation approach, in which 100 randomly selected instances per domain were evaluated without revealing the corresponding system.

However, we still needed an automated heuristic for the filtering step described in Section 4, so we evaluated the effectiveness of LLM judges for this task. Specifically, we adapted Prometheus 2 (Kim et al., 2024) to our evaluation setting. With the prompt available in Appendix A.3, we obtained a substantially higher correlation with manually annotated instances than traditional metrics such as BLEU4, which is used in the original dataset. However, we observed that the LLM-based judge tends to overestimate the performance of the baselines, leading to a higher rate of false positives.

Due to this limitation, the use of Prometheus is limited to filtering during the unsupervised self-improvement pipeline, where, additionally, we were able to check via manual evaluation that models did not overfit to Prometheus preferences. Moreover,

[1] Fashion attributes: assetType, availableSizes, brand, color, customerReview, pattern, prefab_path, price, size, sleeveLength, type.

[2] Furniture attributes: brand, color, customerRating, materials, prefab_path, price, type.

[3] The dialogue intent of the utterance is not used by the model. We use that information for analysis purposes.

it was used as a signal in some experiments, where the necessity for an automatic evaluation method was a must. However, in this cases, we took the necessary precautions to minimize Prometheus' biases by only comparing the same methods.

6 Experimental Results

The systems were evaluated using the blind manual assessment approach described in the previous section, judging the instances as correct or incorrect. The same evaluation instances were maintained across all systems to ensure comparability. The results for each system are shown in Table 1.

System	Fashion accuracy	Furniture accuracy
Blind 70B	30%	25%
All-in-context 70B	50%	77%
ReAct ICL 8B	31%	42%
ReAct ICL 70B	60%	80%
ReAct FT 8B	65%	75%
ReAct FT 70B	**68%**	76%
USI ReAct 8B	67%	**82%**

Table 1: Results using manual evaluation for all systems in both domains.

The results demonstrate that the ReAct 70B systems outperform the baselines, highlighting the effectiveness of the ReAct approach. In general, the fine-tuned models surpass their respective in-context learning counterparts. Notably, ReAct FT 8B achieves remarkable performance, significantly outperforming the ReAct ICL 70B model in the fashion domain (65% vs. 60%), despite having significantly fewer parameters.

Interestingly, USI ReAct 8B also performs competitively against 70B models, surpassing ReAct ICL 70B in both domains and achieving results comparable to ReAct FT 70B in the fashion domain (-1%) while substantially improving performance in the furniture domain (+8%). This outcome is particularly notable given its smaller parameter size, which offers advantages such as reduced computational cost during inference, despite higher training requirements.

As expected, all systems that use metadata perform better in the furniture domain than in the fashion domain, due to its simpler nature. This phenomenon also is present in the performance gap between the All-in-context baseline and ReAct ICL 70B, which is smaller in the furniture domain than in the fashion domain. As discussed in Section 7.3, this difference arises from the higher number of objects present in fashion scenes, which increases task complexity.

7 Analysis

7.1 Intent Analysis

We know that user utterances vary in their reasoning demands (e.g., *"compare the two tables"* vs. *"add this table to the cart"*). To investigate system performance, we leverage the intent annotations provided in the SIMMC 2.1 dataset to evaluate with respect to the user's intent.

As described by Moon et al. (2020), each utterance (both user and system) is annotated with two levels of intent granularity. The higher-level intents, also referred to as *dialogue acts*, fall into three categories: `ASK`, `INFORM`, and `REQUEST`. The `ASK` intent denotes information-seeking utterances, `INFORM` indicates those that provide information, and `REQUEST` corresponds to utterances requesting an action.

Each main intent is further subdivided into finer-grained sub-intents. Among them, the `GET` sub-intent is the most versatile, appearing under all three dialogue acts. It is used for requesting an item or retrieving its attributes. When combined with `REQUEST`, as in `REQUEST:GET`, it typically denotes generic item requests.[4] In contrast, `ASK:GET` focuses on querying metadata information,[5] while `INFORM:GET` involves suggesting new items similar to others, often with additional constraints.[6] The `INFORM:DISAMBIGUATE` sub-intent corresponds to utterances resolving prior ambiguity,[7] whereas `INFORM:REFINE` introduces further restrictions on an existing search.[8] Finally, `REQUEST:ADD_TO_CART` denotes adding an item to the cart,[9] and `REQUEST:COMPARE`

[4] `REQUEST:GET`: *Can you show me some shoes?*

[5] `ASK:GET`: *What's the price of that jacket?*

[6] `INFORM:GET`: *Do you have anything in a similar size to the grey shirt?*

[7] `INFORM:DISAMBIGUATE`: *I meant the grey one.*

[8] `INFORM:REFINE`: *Can we look for more expensive ones.*

[9] `REQUEST:ADD_TO_CART`: *Add the grey hat to my cart.*

	Total	Blind	All-in-context	ReAct ICL 70B	ReAct FT 70B	USI ReAct 8B
ASK:GET	17	6%	41%	**76%**	65%	41%
INFORM:DISAMBIGUATE	16	0%	50%	62%	69%	**88%**
INFORM:GET	17	12%	59%	59%	65%	**77%**
INFORM:REFINE	16	0%	37%	62%	63%	**75%**
REQUEST:ADD_TO_CART	44	57%	59%	68%	84%	**95%**
REQUEST:COMPARE	20	0%	69%	65%	65%	**70%**
REQUEST:GET	70	38%	**80%**	77%	73%	67%

Table 2: Correctly answered utterances filtered by the questions' intent.

indicates comparisons among items.[10]

Table 2 presents system accuracy across different user intents. As expected, the Blind baseline fails to handle questions requiring object-specific information (e.g., REQUEST:COMPARE, INFORM:REFINE). The All-in-context model shows a substantial improvement over this baseline, underscoring the importance of access to external information. It performs particularly well on queries involving multiple objects (e.g., REQUEST:GET, REQUEST:COMPARE). In contrast, ReAct ICL 70B achieves notable gains on single-object reasoning tasks (e.g., ASK:GET) while maintaining accuracy on the remaining intents. ReAct FT 70B improves performance across nearly all intent types and exhibits stronger alignment on simpler tasks relying primarily on contextual understanding (e.g., REQUEST:ADD_TO_CART). These gains are further amplified through the self-improvement loop, with USI ReAct 8B achieving the best overall results across most intents. Interestingly, USI ReAct 8B shows marked improvement in the INFORM category, demonstrating enhanced ability to leverage scene information to identify new relevant objects. However, its accuracy decreases in the ASK:GET intent, suggesting that future iterations of the pipeline should aim to preserve performance on simpler query-based reasoning tasks.

7.2 Evolution of Self-Improving

We assess the effectiveness of the self-improvement approach by tracking the increase in both the number of trajectories rated with a score of 5 and the average validation score assigned by Prometheus. Table 3 illustrates the progressive improvement of the model and the corresponding expansion of the trajectory set across iterations. In terms of the Prometheus score, the model improves from an initial value of 2.77 to 3.67 after five iterations. Likewise, the number of top-rated trajectories (score 5) increases substantially, from 5,492 in the fashion domain (and 2,342 in furniture) at the initial stage to 11,073 in fashion (and 4,408 in furniture) after the final iteration. Note that the increasing ratio is similar in the two domains.

The analysis reveals that the first iteration yields the largest improvement, increasing the number of filtered instances by 55% in the fashion domain and 43% in furniture. Subsequent iterations result in smaller gains, indicating that the initial fine-tuning step plays a crucial role in adapting the models. This trend is consistent with the performance of ReAct FT 8B, as shown in Table 1, where a single fine-tuning iteration leads to substantial performance gains. We stopped training after 5 iterations, as it showed worse performance in the 6^{th} fine-tuning step.

The Prometheus score analysis was conducted exclusively on the validation partition of the fashion domain; thus, we acknowledge that the findings may be limited in scope. Nevertheless, it is worth noting that fashion represents the most challenging domain in our setup, and the observed trends are expected to generalize to the furniture domain as well.

7.3 Effect of Scene Complexity

We analyze how scene complexity, measured by the number of included objects, influences model performance. We investigate whether (1) the higher performance observed in the furniture domain arises from its smaller number of objects, and (2) whether the All-in-context baseline may struggle as the number of objects increases substantially.

We conducted an additional analysis by

[10] REQUEST:COMPARE: *What sizes are available for each?*

Iteration	Initial	1	2	3	4	5	6
Fashion utterances	5492	8542	9653	10289	10789	**11073**	-
Furniture utterances	2342	3368	3893	4117	4206	**4408**	-
Prometheus score (val)	2.77	3.51	3.56	3.59	3.63	**3.67**	3.63

Table 3: Training set evolution throughout the self-improvement process. This table shows the number of high-quality training utterances (Prometheus scoring 5/5) used for each iteration, along with the mean Prometheus score on the validation set. Selected iteration's results in bold.

augmenting the validation set with synthetic (fake) objects in the scene metadata. Multiple dataset variants were created, each containing a different number of total objects per scene, ranging from a minimum of 20 up to 150. For dialogues with fewer objects than the target count, we inserted fake objects into the metadata. These synthetic entries were generated using attribute values not originally present in the scene, such as novel colors, sizes, and clothing categories, to ensure clear differentiation from real objects.

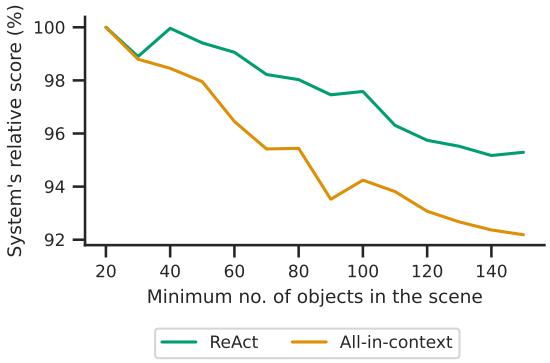

Figure 3: Prometheus average scores for the different systems as the number of objects increases in the fashion domain.

The results of this analysis are presented in Figure 3. It shows the performance gap between ReAct ICL 70B and the All-in-context baseline. Notably, as the number of objects increases, the performance of the All-in-context baseline degrades substantially. In contrast, ReAct ICL 70B exhibits greater robustness under complex conditions. These findings reinforce the effectiveness of the reasoning and tool-use framework in scenarios that require managing wide context windows and highly specific situational information.

7.4 Filtering noisy triplets

Experimental results indicate that Prometheus effectively identifies high-quality reasoning trajectories, enabling smaller fine-tuned models to outperform larger ICL-based models. It is important to note, however, that the LLM evaluates each trajectory holistically, rather than assessing the quality of individual *thought–action–observation* triplets within the reasoning process. This limitation may influence the outcome of fine-tuning, as the model could inadvertently learn to produce longer or inefficient trajectories that ultimately lead to incorrect responses.

We define a simple heuristic that filters out incorrect triplets. The filter discards malformed actions, such as incomplete operations or invalid search queries, and identifies multiple consecutive uses of the `Look[]` tool as inefficient, since they consistently yield identical results.

System	Filtered trajectories	W/o filtering
ReAct FT 70B	**2.8951**(-0.67)	3.5608
ReAct FT 8B	**2.7906**(-0.53)	3.3172
USI ReAct 8B	**2.4974**(-0.29)	2.7849

Table 4: Mean number of triplets generated by each system based on whether triplet filtering was applied. Lower is better.

Table 4 shows that the number of generated triplets decreases substantially when the model is fine-tuned on cleaned trajectories. This reduction is particularly pronounced for models that initially produce a large number of triplets. Interestingly, the non-filtered USI ReAct 8B model generates relatively few triplets, suggesting that the iterative self-learning framework implicitly learns to discard trajectories containing noisy or uninformative triplets.

The results shown in Table 5 highlight the importance of using clean trajectories. Most models trained on the filtered dataset exhibit considerable improvements. This is particu-

System	Filtered trajectories	W/o filtering
ReAct FT 70B	72%	**74%**
ReAct FT 8B	**70%**	62.5%
USI ReAct 8B	**74.5%**	73.5%

Table 5: Results of fine-tuned systems based on whether triplet filtering was applied.

larly pronounced in the ReAct FT 8B model, improving the model up to 7 points.

7.5 Generalization of Fine-tuned ReAct

We evaluate the cross-domain generalization and transferability of our fine-tuning pipeline. This analysis is particularly relevant, as fine-tuned models can easily memorize object properties within a domain, leading to overfitting (Hemanthage et al., 2023). To assess this, we train the models on one domain and evaluate them on both. We further compare our fine-tuned model against a BART state-of-the-art architecture in this task (Lee et al., 2022; Lewis et al., 2020), and use it as a baseline to check for possible memorization differences between the architectures.

System	Trained on	Fashion accuracy	Furniture accuracy
BART	Fashion	**19%**	10%
	Furniture	7%	**15%**
ReAct FT 8B	Fashion	21%	**22%**
	Furniture	20%	**25%**

Table 6: Percentage of instances achieving a Prometheus evaluation score of 3 or higher. The models were trained only on a specific domain and evaluated in both.

The results of these experiments are presented in Table 6. Our fine-tuning approach demonstrates superior cross-domain generalization, as the performance drop observed during cross-domain evaluation is substantially smaller than that of the BART baseline. Interestingly, our model trained on the fashion domain achieves higher scores when evaluated on furniture, consistent with prior observations that domain complexity influences performance. In contrast, the BART models exhibit a pronounced decline in performance when tested on out-of-domain instances.

8 Conclusions

In this work, we demonstrated the successful adaptation of the ReAct framework to task-oriented dialogue systems, mitigating hallucination issues by grounding the model in its environment through dynamic information retrieval. Furthermore, we showed that an effective unsupervised fine-tuning pipeline can be developed to leverage reasoning trajectories without relying on human feedback.

Our experiments on the SIMMC dataset demonstrate that the proposed approach not only outperforms baseline LLMs that rely solely on in-context information, but also exhibits greater robustness in complex scenarios characterized by high object density. Importantly, our self-improvement pipeline enables a Llama 3 8B model to match or even surpass the performance of a Llama 3 70B in-context system. Moreover, our analysis shows that filtering inefficient reasoning steps improves both inference efficiency and overall performance, while fine-tuning on reasoning trajectories fosters strong cross-domain generalization.

9 Limitations

While our approach demonstrates the effective adaptation of the ReAct framework to dialogue systems, several limitations remain, highlighting opportunities for future research.

First, our experiments are conducted exclusively on the SIMMC 2.1 dataset. Although this dataset provides a rich, multimodal environment with multiple domains, extending the evaluation to a broader range of task-oriented dialogue datasets would enable a more comprehensive assessment of the proposed approach's robustness.

Second, despite the multimodal nature of the SIMMC 2.1 dataset, our method utilizes only textual and structured metadata, omitting the direct use of visual input. Integrating visual signals directly into the ReAct framework remains an important direction for future research.

Finally, our experimental results rely primarily on manual evaluation, which limits the assessment to a few hundred examples. This highlights the critical need for developing reliable automatic evaluation methods, which would enable a more scalable analysis of system performance across larger sets of examples.

Acknowledgments

This work was developed under the grant 101135724 (LUMINOUS) of the EU Horizon Europe Framework.

We wish to thank many of our colleagues in the research group for their support, including technical and administrative assistance, throughout this project. A special thank you goes to Oier Ijurco for his valuable insights into the dataset and his participation in shaping ideas behind this work.

References

Acikgoz, E. C., C. Guo, S. Dey, A. Datta, T. Kim, G. Tur, and D. Hakkani-Tur. 2025. TD-EVAL: Revisiting task-oriented dialogue evaluation by combining turn-level precision with dialogue-level comparisons. In F. Béchet, F. Lefèvre, N. Asher, S. Kim, and T. Merlin, editors, *Proceedings of the 26th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 113–132, Avignon, France, August. Association for Computational Linguistics.

Celikyilmaz, A., L. Deng, and D. Hakkani-Tür, 2018. *Deep Learning in Spoken and Text-Based Dialog Systems*, pages 49–78. Springer Singapore, Singapore.

Dhuliawala, S., M. Komeili, J. Xu, R. Raileanu, X. Li, A. Celikyilmaz, and J. Weston. 2023. Chain-of-verification reduces hallucination in large language models.

Elizabeth, M., M. Veyret, M. Couceiro, O. Dusek, and L. M. Rojas Barahona. 2025. Exploring ReAct prompting for task-oriented dialogue: Insights and shortcomings. In M. I. Torres, Y. Matsuda, Z. Callejas, A. del Pozo, and L. F. D'Haro, editors, *Proceedings of the 15th International Workshop on Spoken Dialogue Systems Technology*, pages 143–153, Bilbao, Spain, May. Association for Computational Linguistics.

Eric, M., R. Goel, S. Paul, A. Kumar, A. Sethi, P. Ku, A. K. Goyal, S. Agarwal, S. Gao, and D. Hakkani-Tur. 2019. Multiwoz 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines.

Grattafiori, A., A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, A. Yang, A. Fan, A. Goyal, A. Hartshorn, A. Yang, A. Mitra, A. Sravankumar, A. Korenev, A. Hinsvark, A. Rao, A. Zhang, A. Rodriguez, A. Gregerson, A. Spataru, B. Roziere, B. Biron, B. Tang, B. Chern, C. Caucheteux, C. Nayak, C. Bi, C. Marra, C. McConnell, C. Keller, C. Touret, C. Wu, C. Wong, C. C. Ferrer, C. Nikolaidis, D. Allonsius, D. Song, D. Pintz, D. Livshits, D. Wyatt, D. Esiobu, D. Choudhary, D. Mahajan, D. Garcia-Olano, D. Perino, D. Hupkes, E. Lakomkin, E. AlBadawy, E. Lobanova, E. Dinan, E. M. Smith, F. Radenovic, F. Guzmán, F. Zhang, G. Synnaeve, G. Lee, G. L. Anderson, G. Thattai, G. Nail, G. Mialon, G. Pang, G. Cucurell, H. Nguyen, H. Korevaar, H. Xu, H. Touvron, I. Zarov, I. A. Ibarra, I. Kloumann, I. Misra, I. Evtimov, J. Zhang, J. Copet, J. Lee, J. Geffert, J. Vranes, J. Park, J. Mahadeokar, J. Shah, J. van der Linde, J. Billock, J. Hong, J. Lee, J. Fu, J. Chi, J. Huang, J. Liu, J. Wang, J. Yu, J. Bitton, J. Spisak, J. Park, J. Rocca, J. Johnstun, J. Saxe, J. Jia, K. V. Alwala, K. Prasad, K. Upasani, K. Plawiak, K. Li, K. Heafield, K. Stone, K. El-Arini, K. Iyer, K. Malik, K. Chiu, K. Bhalla, K. Lakhotia, L. Rantala-Yeary, L. van der Maaten, L. Chen, L. Tan, L. Jenkins, L. Martin, L. Madaan, L. Malo, L. Blecher, L. Landzaat, L. de Oliveira, M. Muzzi, M. Pasupuleti, M. Singh, M. Paluri, M. Kardas, M. Tsimpoukelli, M. Oldham, M. Rita, M. Pavlova, M. Kambadur, M. Lewis, M. Si, M. K. Singh, M. Hassan, N. Goyal, N. Torabi, N. Bashlykov, N. Bogoychev, N. Chatterji, N. Zhang, O. Duchenne, O. Çelebi, P. Alrassy, P. Zhang, P. Li, P. Vasic, P. Weng, P. Bhargava, P. Dubal, P. Krishnan, P. S. Koura, P. Xu, Q. He, Q. Dong, R. Srinivasan, R. Ganapathy, R. Calderer, R. S. Cabral, R. Stojnic, R. Raileanu, R. Maheswari, R. Girdhar, R. Patel, R. Sauvestre, R. Polidoro, R. Sumbaly, R. Taylor, R. Silva, R. Hou, R. Wang, S. Hosseini, S. Chennabasappa, S. Singh, S. Bell, S. S. Kim, S. Edunov, S. Nie, S. Narang, S. Raparthy, S. Shen, S. Wan, S. Bhosale, S. Zhang, S. Vandenhende, S. Batra, S. Whitman, S. Sootla, S. Col-

lot, S. Gururangan, S. Borodinsky, T. Herman, T. Fowler, T. Sheasha, T. Georgiou, T. Scialom, T. Speckbacher, T. Mihaylov, T. Xiao, U. Karn, V. Goswami, V. Gupta, V. Ramanathan, V. Kerkez, V. Gonguet, V. Do, V. Vogeti, V. Albiero, V. Petrovic, W. Chu, W. Xiong, W. Fu, W. Meers, X. Martinet, X. Wang, X. Wang, X. E. Tan, X. Xia, X. Xie, X. Jia, X. Wang, Y. Goldschlag, Y. Gaur, Y. Babaei, Y. Wen, Y. Song, Y. Zhang, Y. Li, Y. Mao, Z. D. Coudert, Z. Yan, Z. Chen, Z. Papakipos, A. Singh, A. Srivastava, A. Jain, A. Kelsey, A. Shajnfeld, A. Gangidi, A. Victoria, A. Goldstand, A. Menon, A. Sharma, A. Boesenberg, A. Baevski, A. Feinstein, A. Kallet, A. Sangani, A. Teo, A. Yunus, A. Lupu, A. Alvarado, A. Caples, A. Gu, A. Ho, A. Poulton, A. Ryan, A. Ramchandani, A. Dong, A. Franco, A. Goyal, A. Saraf, A. Chowdhury, A. Gabriel, A. Bharambe, A. Eisenman, A. Yazdan, B. James, B. Maurer, B. Leonhardi, B. Huang, B. Loyd, B. D. Paola, B. Paranjape, B. Liu, B. Wu, B. Ni, B. Hancock, B. Wasti, B. Spence, B. Stojkovic, B. Gamido, B. Montalvo, C. Parker, C. Burton, C. Mejia, C. Liu, C. Wang, C. Kim, C. Zhou, C. Hu, C.-H. Chu, C. Cai, C. Tindal, C. Feichtenhofer, C. Gao, D. Civin, D. Beaty, D. Kreymer, D. Li, D. Adkins, D. Xu, D. Testuggine, D. David, D. Parikh, D. Liskovich, D. Foss, D. Wang, D. Le, D. Holland, E. Dowling, E. Jamil, E. Montgomery, E. Presani, E. Hahn, E. Wood, E.-T. Le, E. Brinkman, E. Arcaute, E. Dunbar, E. Smothers, F. Sun, F. Kreuk, F. Tian, F. Kokkinos, F. Ozgenel, F. Caggioni, F. Kanayet, F. Seide, G. M. Florez, G. Schwarz, G. Badeer, G. Swee, G. Halpern, G. Herman, G. Sizov, Guangyi, Zhang, G. Lakshminarayanan, H. Inan, H. Shojanazeri, H. Zou, H. Wang, H. Zha, H. Habeeb, H. Rudolph, H. Suk, H. Aspegren, H. Goldman, H. Zhan, I. Damlaj, I. Molybog, I. Tufanov, I. Leontiadis, I.-E. Veliche, I. Gat, J. Weissman, J. Geboski, J. Kohli, J. Lam, J. Asher, J.-B. Gaya, J. Marcus, J. Tang, J. Chan, J. Zhen, J. Reizenstein, J. Teboul, J. Zhong, J. Jin, J. Yang, J. Cummings, J. Carvill, J. Shepard, J. McPhie, J. Torres, J. Ginsburg, J. Wang, K. Wu, K. H. U, K. Saxena, K. Khandelwal, K. Zand, K. Matosich, K. Veeraraghavan, K. Michelena, K. Li, K. Jagadeesh, K. Huang, K. Chawla, K. Huang, L. Chen, L. Garg, L. A, L. Silva, L. Bell, L. Zhang, L. Guo, L. Yu, L. Moshkovich, L. Wehrstedt, M. Khabsa, M. Avalani, M. Bhatt, M. Mankus, M. Hasson, M. Lennie, M. Reso, M. Groshev, M. Naumov, M. Lathi, M. Keneally, M. Liu, M. L. Seltzer, M. Valko, M. Restrepo, M. Patel, M. Vyatskov, M. Samvelyan, M. Clark, M. Macey, M. Wang, M. J. Hermoso, M. Metanat, M. Rastegari, M. Bansal, N. Santhanam, N. Parks, N. White, N. Bawa, N. Singhal, N. Egebo, N. Usunier, N. Mehta, N. P. Laptev, N. Dong, N. Cheng, O. Chernoguz, O. Hart, O. Salpekar, O. Kalinli, P. Kent, P. Parekh, P. Saab, P. Balaji, P. Rittner, P. Bontrager, P. Roux, P. Dollar, P. Zvyagina, P. Ratanchandani, P. Yuvraj, Q. Liang, R. Alao, R. Rodriguez, R. Ayub, R. Murthy, R. Nayani, R. Mitra, R. Parthasarathy, R. Li, R. Hogan, R. Battey, R. Wang, R. Howes, R. Rinott, S. Mehta, S. Siby, S. J. Bondu, S. Datta, S. Chugh, S. Hunt, S. Dhillon, S. Sidorov, S. Pan, S. Mahajan, S. Verma, S. Yamamoto, S. Ramaswamy, S. Lindsay, S. Lindsay, S. Feng, S. Lin, S. C. Zha, S. Patil, S. Shankar, S. Zhang, S. Zhang, S. Wang, S. Agarwal, S. Sajuyigbe, S. Chintala, S. Max, S. Chen, S. Kehoe, S. Satterfield, S. Govindaprasad, S. Gupta, S. Deng, S. Cho, S. Virk, S. Subramanian, S. Choudhury, S. Goldman, T. Remez, T. Glaser, T. Best, T. Koehler, T. Robinson, T. Li, T. Zhang, T. Matthews, T. Chou, T. Shaked, V. Vontimitta, V. Ajayi, V. Montanez, V. Mohan, V. S. Kumar, V. Mangla, V. Ionescu, V. Poenaru, V. T. Mihailescu, V. Ivanov, W. Li, W. Wang, W. Jiang, W. Bouaziz, W. Constable, X. Tang, X. Wu, X. Wang, X. Wu, X. Gao, Y. Kleinman, Y. Chen, Y. Hu, Y. Jia, Y. Qi, Y. Li, Y. Zhang, Y. Zhang, Y. Adi, Y. Nam, Yu, Wang, Y. Zhao, Y. Hao, Y. Qian, Y. Li, Y. He, Z. Rait, Z. DeVito, Z. Rosnbrick, Z. Wen, Z. Yang, Z. Zhao, and Z. Ma. 2024. The llama 3 herd of models.

Gu, J., X. Jiang, Z. Shi, H. Tan, X. Zhai, C. Xu, W. Li, Y. Shen, S. Ma, H. Liu,

S. Wang, K. Zhang, Y. Wang, W. Gao, L. Ni, and J. Guo. 2025. A survey on llm-as-a-judge.

Gupta, A., A. Ravichandran, Z. Zhang, S. Shah, A. Beniwal, and N. Sadagopan. 2024. Dard: A multi-agent approach for task-oriented dialog systems.

Hemanthage, B., C. Dondrup, P. Bartie, and O. Lemon. 2023. SimpleMTOD: A simple language model for multimodal task-oriented dialogue with symbolic scene representation. In M. Amblard and E. Breitholtz, editors, *Proceedings of the 15th International Conference on Computational Semantics*, pages 293–304, Nancy, France, June. Association for Computational Linguistics.

Hu, E. J., Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. 2021. Lora: Low-rank adaptation of large language models.

Kim, S., J. Suk, S. Longpre, B. Y. Lin, J. Shin, S. Welleck, G. Neubig, M. Lee, K. Lee, and M. Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models.

Kottur, S. and S. Moon. 2023. Overview of situated and interactive multimodal conversations (SIMMC) 2.1 track at DSTC 11. In Y.-N. Chen, P. Crook, M. Galley, S. Ghazarian, C. Gunasekara, R. Gupta, B. Hedayatnia, S. Kottur, S. Moon, and C. Zhang, editors, *Proceedings of the Eleventh Dialog System Technology Challenge*, pages 235–241, Prague, Czech Republic, September. Association for Computational Linguistics.

Kottur, S., S. Moon, A. Geramifard, and B. Damavandi. 2021. SIMMC 2.0: A task-oriented dialog dataset for immersive multimodal conversations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4903–4912, Online and Punta Cana, Dominican Republic, November. Association for Computational Linguistics.

Lee, H., O. J. Kwon, Y. Choi, M. Park, R. Han, Y. Kim, J. Kim, Y. Lee, H. Shin, K. Lee, and K.-E. Kim. 2022. Learning to embed multi-modal contexts for situated conversational agents. In M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz, editors, *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 813–830, Seattle, United States, July. Association for Computational Linguistics.

Lewis, M., Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July. Association for Computational Linguistics.

Lin, C.-Y. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July. Association for Computational Linguistics.

Liu, C.-W., R. Lowe, I. Serban, M. Noseworthy, L. Charlin, and J. Pineau. 2016. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In J. Su, K. Duh, and X. Carreras, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132, Austin, Texas, November. Association for Computational Linguistics.

Moon, S., S. Kottur, P. A. Crook, A. De, S. Poddar, T. Levin, D. Whitney, D. Difranco, A. Beirami, E. Cho, R. Subba, and A. Geramifard. 2020. Situated and interactive multimodal conversations.

Papineni, K., S. Roukos, T. Ward, and W.-J. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In P. Isabelle, E. Charniak, and D. Lin, editors, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Wang, H., L. Wang, Y. Du, L. Chen, J. Zhou, Y. Wang, and K.-F. Wong. 2025. A survey of the evolution of language model-based dialogue systems: Data, task and models.

Wei, J., X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Xu, H.-D., X.-L. Mao, P. Yang, F. Sun, and H. Huang. 2024. Rethinking task-oriented dialogue systems: From complex modularity to zero-shot autonomous agent. In L.-W. Ku, A. Martins, and V. Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2748–2763, Bangkok, Thailand, August. Association for Computational Linguistics.

Yao, S., J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao. 2023. React: Synergizing reasoning and acting in language models.

Yi, Z., J. Ouyang, Z. Xu, Y. Liu, T. Liao, H. Luo, and Y. Shen. 2025. A survey on recent advances in llm-based multi-turn dialogue systems.

A Prompts

A.1 Baselines prompts

Figure 4: Blind baseline prompt.

You are a shop assistant that must answer the user's question. Give your answer in a single sentence.

Conversation: {Previous turns and current user question}
System:

Figure 5: All-in-context baseline prompt.

You are a shop assistant that must answer the user's question. Give your answer in a single sentence.

Available items' visual features:
{Visual metadata of all objects in the scene}

Available items' hidden features:
{Hidden metadata of all objects in the scene}

Conversation: {Previous turns and current user question}
System:

A.2 ReAct systems prompts

Figure 6: ReAct ICL 70B fashion prompt.

You are a seller in a fashion shop. A client comes to you and you must answer their questions. Solve the questions answering task with interleaving Thought, Action, Observation steps. Thought can reason about the current situation, and Action can be three types:
(1) Search[query], which searches the existing catalogue. You may search for customerReview, availableSizes, brand, price, size or prefab_path. You MUST NOT use other parameters or the query will error. You may use the following operators: ==, ¡=, ¡, ¿=, ¿, !=, in, and, or.
(2) Look[], which returns the items that are visible from your perspective. The position, assetType, color, pattern, sleeveLength, type and prefab_path will become apparent to you.
(3) Finish[answer], which returns the answer to the client and finishes the task.
Here are some examples (DO NOT use information from example conversations in other conversations; you must ask again for the information):
{Few-shot examples}

Figure 7: ReAct ICL 70B furniture prompt.

You are a seller in a furniture shop. A client comes to you and you must answer their questions. Solve the questions answering task with interleaving Thought, Action, Observation steps. Thought can reason about the current situation, and Action can be three types:
(1) Search[query], which searches the existing catalogue. You may search for prefab_path, brand, price, customerRating or materials. You MUST NOT use other parameters or the query will

error. You may use the following operators: ==, ¡=, ¡, ¿=, ¿, !=, and, or.
(2) Look[], which returns the items that are visible from your perspective. The position, corner_pos, type, color and prefab_path will become apparent to you.
(3) Finish[answer], which returns the answer to the client and finishes the task.
Here are some examples (DO NOT use information from example conversations in other conversations; you must ask again for the information):
{Few-shot examples}

A.3 Prometheus Evaluation prompt

Figure 8: Prometheus Evaluation prompt.

###Task Description:
An instruction (might include an Input inside it), a response to evaluate, a reference answer that gets a score of 5, and a score rubric representing a evaluation criteria are given.
1. Write a detailed feedback that assess the quality of the response strictly based on the given score rubric, not evaluating in general.
2. After writing a feedback, write a score that is an integer between 1 and 5. You should refer to the score rubric.
3. The output format should look as follows: "(write a feedback for criteria) [RESULT] (an integer number between 1 and 5)"
4. Please do not generate any other opening, closing, and explanations.
###The instruction to evaluate:
Relevant objects:
{objects}
Conversation:
{question}
###Response to evaluate:
{final_answer}
###Reference Answer (Score 5):
{correct_answer}
###Score Rubrics:
[Does the model correctly represent the relevant object{plural} factually?]
Score 1: The model answers with incorrect information or information about objects that are not relevant.
Score 2:
Score 3:
Score 4:
Score 5: The model answers the question and provides accurate information about the relevant object(s).
###Feedback: