ChatShop: Interactive Information Seeking with Language Agents

Anonymous ACL submission

Abstract

The desire and ability to seek new information strategically are fundamental to human learning but often overlooked in current language agent development. Using a web shopping task as an example, we show that it can be reformulated and solved as a retrieval task without a requirement of interactive information seeking. We then redesign the task to introduce a new role of shopper, serving as a realistically constrained communication channel. The agents in our proposed ChatShop task explore user preferences in open-ended conversation to make informed decisions. Our experiments demonstrate that the proposed task can effectively evaluate the agent's ability to explore and gradually accumulate information through multi-turn interaction. We also show that LLM-simulated shoppers serve as a good proxy to real human shoppers and discover similar error patterns of agents.

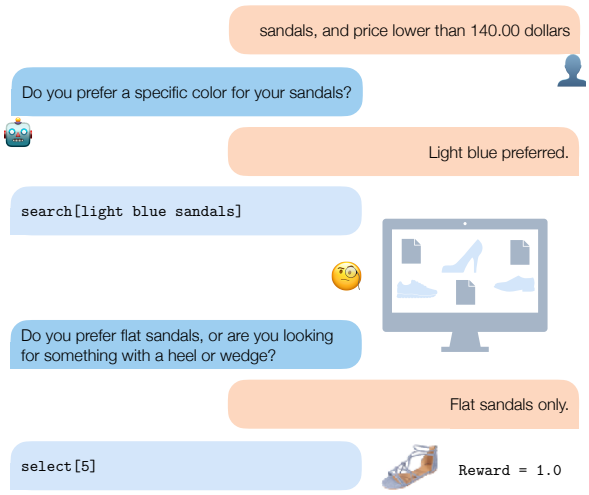

Figure 1: An illustration of the ChatShop task. In contrast to WebShop's original detailed instruction *a non-slip sandals for my wife that is blue in color, 5.5 size*, we provide only the product type (price) and require the agent to narrow down the search space by interacting with the shopper and product database.

1 Introduction

Recent studies have explored Large Language Models (LLMs) as autonomous agents in general problem solving (Zhou et al., 2023; Liu et al., 2023b; Xie et al., 2023). In their design, the component of information seeking is often against a static information source such as a knowledge graph or a pile of web documents. The unconstrained access to the information source reduces *interactivity*—the agent does not need to strategically seek new information from the user and its decision-making process is not informed by tracking accumulated information. To investigate this issue, we first examine Webshop (Yao et al., 2022), which evaluates an agent's ability to identify the correct product in a web shopping interface based on user instructions. For this task, we show that since the shopper provides full details of the target product in their instruction, a retrieval system can directly score the relevance of each product to locate the target.

In a realistic scenario, a shopper would start with partial information of the target product which would become clearer after seeing various options the agent might find. The key challenge with designing such a setup is that interactions between the agent and the shopper would require a human-in-the-loop environment, hindering scalable evaluation. Given the strong performance of recent LLM agents, we hypothesize that LLMs themselves would be capable of simulating humans in an interactive web shopping experience (Li et al., 2023b). To test this hypothesis, we repurpose WebShop to propose ChatShop, in which the agent starts with an unspecific goal instruction—only the coarse type of product. The lack of specificity in the instruction creates a challenge of task ambiguity (Tamkin et al., 2023), which can only be resolved by effectively gathering information from the shopper and the website environment about

products (Figure 1). The challenge is amplified by other inherent complexities such as searching the vast product space and tool usage.

We benchmark a range of agents with both GPT-3.5/4 and a Llama 2 variant as base models in environments where the role of the shopper is played by humans or LLMs. Experimental results verify the challenges introduced by the information need. We further evaluate how good an LLM at simulating the interaction with real human shoppers in a human study. The benchmarking results and the failure patterns show that the LLM simulated environment is as effective in recovering the gap between agents. We hope our work can drive the automatic evaluation of language agents towards more complex and meaningful interactions with (simulated) humans.[1]

2 Related Work

Information Seeking Tasks Language agents' information-seeking ability has long been a focus of AI research, especially in the context of question answering and task-oriented dialogue (Bachman et al., 2016; Dhingra et al., 2017; Zamani et al., 2022; Zhou et al., 2023). In such tasks, the agent usually receives an information need from the user and accesses external knowledge sources to gather information, a task which can often be formulated as a single-turn retrieval problem. The constraints of such interaction are often artificial (Yuan et al., 2020). In contrast, the constraints in ChatShop task originate from a realistic situation of a human party in a web shopping scenario.

Human-AI Collaboration More recently, there has been a growing interest in studying human-AI collaboration via LLMs. MINT (Wang et al., 2023) benchmarks a range of LLM agents in leveraging human or AI-simulated feedback to improve multi-turn problem solving. Unlike ChatShop, this feedback can be viewed as a form of natural language supervision, which is beneficial but not required to solve the task. DialOp (Lin et al., 2023) focuses on the agent's ability of planning based on human preferences in a grounded dialogue setting. Compared to ChatShop, the tasks in DialOp has a narrower and synthetic search space. Li et al. (2023a) propose a learning framework for LLMs to elicit human preferences in tasks such as content recommendation, however, their tasks focus on exploration guided by the general world knowledge stored in the LLM weights internally, whereas in ChatShop, the exploration is grounded in an external real-world product space.

3 ChatShop

This section starts with a review of the WebShop task and then describes the design and evaluation protocol of our repurposed ChatShop task.

WebShop and Retrieval Solution In WebShop, an agent is given a goal instruction and navigates a website to identify the correct product from more than a million candidates. The performance of the agent is evaluated on the reward calculated from the final product selection. The reward function is based on the title string similarity and attribute coverage of the selected product compared to the goal product. The WebShop task can be formulated as retrieval problem. Each product represented by a textual description can be ranked based on its relevance to the goal instruction. We fine-tune a BERT-based model and achieves 87.2 average rewards in evaluation (Appendix A), which surpasses the reported human expert's rewards of 82.1.

Agent and Shopper The proposed ChatShop task involves two roles: a shopper with the intent to purchase an item and an agent that assists the shopper in finding the correct product. In our evaluation of information-seeking capabilities, the shopper, as the primary source of information, has access to the target item. It is either played by a real human or simulated by a language model with a fixed setup. On the other hand, a variety of language agents can be developed and benchmarked in the agent role.

Goal Instruction In ChatShop, we aim to create a starting point with limited information for the agent to explore and accumulate information. We achieve this by simplifying the goal instructions of WebShop to a basic description of the type of item, hiding all attributes and options of the target product, pending the agent's proactive discovery. We process the 1500 goal instructions in the dev and test sets of Webshop and obtain the simplified instructions using GPT-3.5 and few-shot prompts. The simplified instructions are six times shorter and have fewer unique tokens than the original instructions, which suggests a greater degree of task ambiguity.[2]

[1] Data and code to be shared upon publication.

[2] See Appendix D for corpus statistics (Table 4) and actual prompts used.

Action Space Three actions are available to the agent: 1) `search[query]`: initiate a search to a BM25 search engine, which returns a ranked list of products; 2) `select[index]`: when a single product is determined, the agent can finalize its recommendation; 3) `question[content]`: when more information is needed for a precise decision, the agent can interact with the shopper for further clarification.

Communication Channel In the task, we investigate two types of interaction. 1) open-ended text-based interaction: the agent is allowed to ask open-ended questions and the shopper responds naturally in text. 2) instance-based comparison: the agent presents an item to the shopper, in return the shopper provides comments on the item by comparing it to the requirements of the target product. Since the shopper has knowledge of the exact target product, there is a risk of the shopper directly revealing the target product through any communication channels. To prevent this, we limit the length of the shopper's response and employ a few other techniques.

Limit and Reward We do not put any limit on the tool usage, but we limit the maximum number of questions the agent can ask in each session. At the end of the session, when a single product is selected, the same reward is calculated as in the WebShop task. The primary challenge here is for the agents to develop a structured understanding of the product space to identify plausible, distinguishable features and use this understanding to effectively communicate with the shopper.

4 Experiments

We use OpenAI's GPT-3.5 to simulate the shopper for automatic evaluation. The simulated shopper is provided with the product title, the required attributes, and options of the target product. The shopper is instructed to respond to the agent's questions using fewer than 5 words and with a token limit of 10. We allocate a question budget of 5 for each session. Unless specified otherwise, we assess the agent over 100 sessions. In practice, we observe that the agent's performance remains consistent with this number of examples.

4.1 Benchmarking Agents in ChatShop

We select three representative LLMs (OpenAI's GPT-3.5/4 and CODELLAMA-32b) as the backbone of the agents in our study.[3] The complexity of this multi-turn task and the constrained context length of the LLMs make it impractical to include few-shot demonstrations in prompts. We thus carefully design zero-shot prompts and a conversation history compression strategy to instruct the agent to reason and generate valid actions situationally.

	CodeLlama	GPT-3.5	GPT-4
None	34.3	43.4	48.8
Open-ended	-	40.6	49.7
Instance	-	40.4	51.3
Full Info	64.5	76.0	80.1

Table 1: Avg. rewards of (*auto q*) agents under different settings of information disclosure. CODELLAMA cannot perform under the interactive settings without advanced prompting strategies.

We implement three prompting strategies with action enforcing: 1) *auto q*: the agent decides in its own whether to ask questions or search up to a point it chooses to finalize the task with a product selection; 2) *all q*: the agent does a search at the beginning and asks all possible questions until the budget is used up, then finalizes the task with a product selection; 3) *interleave*: the agent asks questions and searches in an interleaved manner using all the questioning budget. This is designed to greedily utilize the tool usage and questions.

Challenge of Information Scarcity In the results of Table 1, we find that state-of-the-art LLMs can generally achieve a high reward with access to full information in instructions, which mimics the setting of WebShop. However, all of the tested LLM agents perform significantly worse when the information becomes scarce, with a performance drop of more than absolute 30% in average rewards. Moreover, even when given access to interact with a simulated shopper, the agents still struggle to utilize the communication channel effectively, resulting in a performance similar or even lower than the no-interaction setting. We find that basic forms of prompting strategy is inadequate to incentivize the agents to interact with the environment. The agents often feel confident in making decisions based on partial information from the instruction or a few interactions with the shopper, despite being prompted to ask questions until "*the user's criteria clearly match a single product*".

[3] gpt-3.5-turbo-1106 and gpt-4-1106-preview versions.

Strategy	GPT-3.5		GPT-4	
CoT	w/o	w/	w/o	w/
no q	43.4	45.6	48.8	47.5
auto q	40.6	62.7	49.7	59.2
all q	63.7	61.3	63.0	66.3
interleave	64.3	**68.2**	60.5	68.1

Table 2: Avg. rewards of agents with different strategies and the open-ended communication channel. *no q* is the non-interactive baselines. See Appendix C for the instance-based communication channel results.

Advanced Prompting Strategy LLM agents have been shown as incapable to leverage the communication channel in the *auto q* setting. We are interested in whether stronger agents can be achieved by task heuristics and common prompt engineering techniques such as chain-of-thought (CoT) (Wei et al., 2022). For CoT, the agent is instructed to summarize the information gathered up to the current turn and reason about the next action based on the summary. In Table 2, we see that CoT is much more helpful in interactive settings, especially in the *auto q* setting where the agent is otherwise confident and reluctant to ask questions. In the best setting, GPT-3.5 surprisingly outperforms GPT-4, suggesting that stronger base model performance does not always translate to information-seeking task. Although advanced prompting strategies further incentivize the agents, the gap between the best agent and the no-interactive full information baseline remains significant.

4.2 LLM versus Human Shopper

To understand the effectiveness of using LLMs as a reasonable proxy for simulating real human buyer interaction, we compare the performance of the LLM agents with the simulated shopper to that with real human. We recruit 8 participants to play the role of the shopper in the human study. Each participant is asked to complete 10-20 sessions of the ChatShop task. The average completion time for one session is 2.5 minutes.[4] We compare two OpenAI agents in the study, both with the *interleave* strategy. In addition, we allow the GPT-3.5 agent to use CoT style reasoning. We collect in total 100 sessions of human shopping data.

From the results in Table 3, we find that the LLM agents performance with the simulated shopper and the human shopper are consistent. Both environments present similar challenges to the agents and reveal the gap between the two agents.

[4]OpenAI API wait time accounts for about 30% of the total time.

	GPT-3.5	GPT-4
Simulated	59.0	62.8
Human	58.2	63.4

Table 3: Avg. rewards of LLM agents with simulated and human shoppers over 50 sessions.

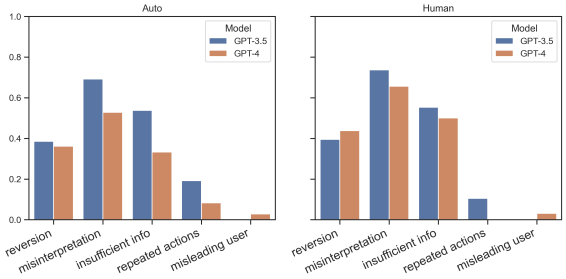

Figure 2: Relative frequency of error types in the LLM agents' failed trajectories with simulated and human shoppers.

Besides the quantitative comparison, we also investigate qualtitatively whether the LLM agents exibit similar failure patterns in both settings. To do this, we first analyze failed trajectories and categorize the failure patterns into five types, associated with systematic limitation of LLMs (Appendix C). Manually going over the lengthy trajectories of the LLM agents can be time-consuming and error-prone. We thus adopt an automatic evaluation method by prompting GPT-4 to tag failed trajectories with the likely causes of failure as a multi-label classification problem. We manually verify a small subset of the model's predictions and find them consistent with our judgement. We then compare the distribution of the failure patterns between the two environments and find them consistent with each other (Figure 2). The inferior GPT-3.5 agent has a higher rate on major error types, the gap is even more pronounced in the simulated shopper environment. The first three error types are widespread, indicating current LLM agents' lack of strategic information seeking and robust long context modeling. The occurrence of the *misleading user* error is rare in both environments.

5 Conclusion

ChatShop presents a information-seeking centric evaluation of language agents, revealing a range of limitations of current LLM models. We hope our fully automatic evaluation pipeline and baseline agents can benefit future exploration.

6 Limitations

Our ChatShop task is realistic in the vast real product space and the interaction with the shopper, but it is still a simplified version of the real-world web shopping scenario. One unrealistic assumption is that the target product is known to the shopper. Relexing this assumption would require real world data on the shopper's knowledge of the target product. Under our current evaluation protocol, the agents are evaluated based on end task performance under a fixed budget of questions. Therefore, it does not capture the quality of interactions for successful sessions as they all receive full rewards. Future work can explore a dynamic budget allocation strategy based on the difficulty of individual sessions or a penalty for asking uninformative questions.

References

Philip Bachman, Alessandro Sordoni, and Adam Trischler. 2016. Towards information-seeking agents. *arXiv preprint arXiv:1612.02605*.

Bhuwan Dhingra, Lihong Li, Xiujun Li, Jianfeng Gao, Yun-Nung Chen, Faisal Ahmed, and Li Deng. 2017. Towards end-to-end reinforcement learning of dialogue agents for information access. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–495, Vancouver, Canada. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Belinda Z. Li, Alex Tamkin, Noah Goodman, and Jacob Andreas. 2023a. Eliciting human preferences with language models. *arXiv preprint arXiv: 2310.11589*.

Yuan Li, Yixuan Zhang, and Lichao Sun. 2023b. Metaagents: Simulating interactions of human behaviors for llm-based task-oriented coordination via collaborative generative agents. *arXiv preprint arXiv:2310.06500*.

Jessy Lin, Nicholas Tomlin, Jacob Andreas, and Jason Eisner. 2023. Decision-oriented dialogue for human-ai collaboration. *arXiv preprint arXiv: 2305.20076*.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023a. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv: 2307.03172*.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. 2023b. Agentbench: Evaluating llms as agents. *arXiv preprint arXiv: 2308.03688*.

Alex Tamkin, Kunal Handa, Avash Shrestha, and Noah D. Goodman. 2023. Task ambiguity in humans and language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Xingyao Wang, Zihan Wang, Jiateng Liu, Yangyi Chen, Lifan Yuan, Hao Peng, and Heng Ji. 2023. Mint: Evaluating llms in multi-turn interaction with tools and language feedback. *arXiv preprint arXiv: 2309.10691*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, E. Chi, F. Xia, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Neural Information Processing Systems*.

Tianbao Xie, Fan Zhou, Zhoujun Cheng, Peng Shi, Luoxuan Weng, Yitao Liu, Toh Jing Hua, Junning Zhao, Qian Liu, Che Liu, Leo Z. Liu, Yiheng Xu, Hongjin Su, Dongchan Shin, Caiming Xiong, and Tao Yu. 2023. Openagents: An open platform for language agents in the wild. *arXiv preprint arXiv: 2310.10634*.

Shunyu Yao, Howard Chen, John Yang, and Karthik R Narasimhan. 2022. Webshop: Towards scalable real-world web interaction with grounded language agents. In *Advances in Neural Information Processing Systems*.

Xingdi Yuan, Jie Fu, Marc-Alexandre Côté, Yi Tay, Chris Pal, and Adam Trischler. 2020. Interactive machine comprehension with information seeking agents. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2325–2338, Online. Association for Computational Linguistics.

Hamed Zamani, Johanne R Trippas, Jeff Dalton, and Filip Radlinski. 2022. Conversational information seeking. *arXiv preprint arXiv:2201.08808*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. 2023. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv: 2307.13854*.

A Retrieval Approach for Webshop

In this section, we review the background of the original WebShop task. We then demonstrate that its interaction challenge is artificial and that a small ranking model can largely solve the task.

WebShop presents a web shopping scenario in which an agent is given a goal instruction (e.g., *I want a noise cancelling cosycost usb microphone*) and navigates a web interface to identify the correct product from more than a million candidates scraped from Amazon. The typical actions available in WebShop involve querying a BM25 search engine, clicking into product details, and confirming a product with corresponding options specified in the goal instruction. The task emphasizes the challenge of recognizing product types, extracting common bi-gram attributes from lengthy product description, matching options and price from a vast collection of products. WebShop has designed a reward function based on the title string similarity and attribute coverage of the selected product compared to the goal product. The performance of the agent is evaluated based on the reward of the final product selected and the success rate of finding a correct product (i.e., reward equals to 1).

The instruction is the only specification of the task and is meant to be sufficiently informative for an agent to identify the correct product. Therefore, we hypothesize that the relevance of each product can be independently determined by the goal instruction alone. As evidence of this hypothesis, we find that, using the instruction as the search query, the built-in BM25 search engine returns a list of 50 products that contains a successful product 86.8% of the time. This finding largely voids the need for the agent to learn how to use the search engine as a tool and diminishes the challenge of large product space exploration.

We further validate this hypothesis by training a simple BERT-based ranking model on the list of products retrieved using the goal instruction. This model applies a cross-attention mechanism between the goal instruction and concatenated textual product information. It uses a margin loss to effectively distinguish suitable from unsuitable products.

Using the retrieval approach, we achieve a 78.3% success rate and 87.2 average rewards on the dev set, which is superior to the reported 59.6% success rate and 82.1 average rewards of human expert annotator (Yao et al., 2022).[5] This result suggests that the task is not challenging in terms of critical interaction that requires strategic planning, but rather that it is associated with the complexity of the instruction and the ambiguity of the task. This observation motivates us to design a new task that focuses on the interaction with the buyer and the website data, rather than the website interface.

	WebShop	ChatShop
# Vocab	2871	1166
Avg. Length	15.1	2.3

Table 4: Corpus statistics of the original and simplified goal instructions. We tokenize the sentences using the nltk library and ignore the stopword tokens in vocabulary counting.

B Experimental Details

B.1 Data Preparation

We use the GPT-3.5 model to extract the coarse product type from the original WebShop goal instructions.[6] The corpus statistics of the 1,500 (1,000 test, 5,00 dev) original and simplified goal instructions are shown in Table 4. We maintain the same training, development, and test splits as defined in the WebShop task. As the agents presented in this study do not require training, we only evaluate and report their performance on the first 100 examples from dev set.

B.2 Prompt Engineering for Agent

We access the OpenAI models via paid APIs. We host CODELLAMA with Hugging Face's transformers library and query via the fastchat (Zheng et al., 2023) API wrapper. These three models are claimed to support long input contexts of at least 16k tokens.[7]

In the zero-shot prompts, the agents see a brief description of the task and the goal instruction, and are instructed to ask questions to the shopper to gather information about the target product until

[5] We do not consider the option selection component of WebShop in this retrieval study, as it merely requires exact string matching.

[6] The WebShop dataset is MIT licensed and can be accessed at https://github.com/princeton-nlp/WebShop.

[7] We find that the 4,096 tokens context length of many LLMs (e.g., gpt-3.5-turbo-0613 and LLAMA 2) is not sufficient for the task.

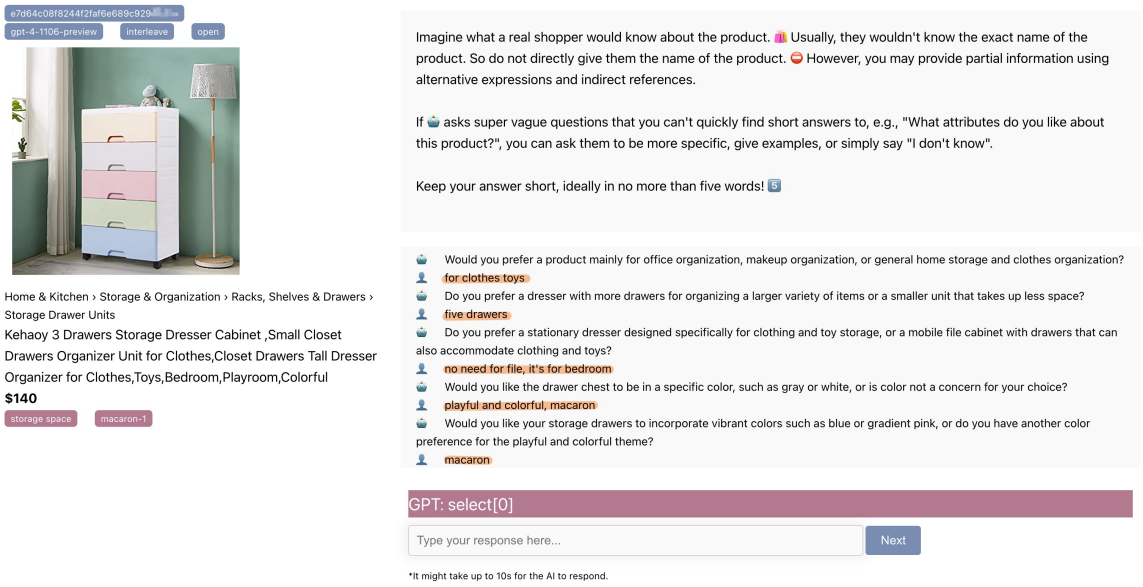

Figure 3: A GPT-4 agent helps a human shopper in the ChatShop task. Picture shows the web interface we build for human evaluation. The left panel provides shopper-related information such as the target product details. The right panel includes the goal instruction and a chat agent interface. The agent can ask questions to the buyer to gather information about the target product. The buyer is asked to answer within a certain length, thus limiting the information transmitted in a single interaction turn.

the shopper's criteria clearly match a single product. Through the conversation, the agents may choose to search the product space using the BM25 search engine. A list of products is returned and the cancatenation of the product titles, attributes, and available options is presented to the agent. For OpenAI models, we provide 20 products for each search action, and for CODELLAMA, we provide 5 products because slower response time and inferior performance in long context modeling.

As the cost and speed of modern search engine are highly optimized, we do not constrain or penalize the use of search engine in ChatShop. However, repeated search actions can lead to a lengthy context for the agent to process and risk exceeding the context window limit. We thus compress the conversation history by hiding the product candidate results from previous search actions, so that the agent can focus on the current state of the game. The old search results are removed from the context, hidden from the agent when a new search is initiated.

For OpenAI models, we use `tools` to mitigate the challenges in enforcing valid agent actions and parsing model generation. For CODELLAMA, we use textual prompts only and interactively add lexical hints (e.g., "What your next search | select would be") to enforce valid actions. We use regular expressions to parse the agent's response. We set the maximum length of the agent's response to 100 tokens, and sample responses until a parseable response is generated.

B.3 Prompt Engineering for Shopper

We use OpenAI's GPT-3.5 to simulate the shopper in the ChatShop task.[8] The shopper is given the product title, the required attributes and options of the target product. To prevent the simulated shopper from directly revealing the target product, we adopt a simple yet effective strategy—limiting the length of the shopper's response. We instruct the shopper to answer the agent's questions in less than 5 words and also cut off the shopper's response if it exceeds 10 tokens. This fixed length constraint also encourages the agent to ask questions in a more strategic manner, optimizing the information density of each interaction turn. We additionally instruct the shopper to avoid revealing the target product voluntarily, but to provide answers in a way that is consistent with the target product. In their response, the shopper reminds the agent the left budget of questions and rejects the agent's request if the budget is exceeded.

[8] We use the `gpt-3.5-turbo-0613` version.

B.4 Sample Conversation

Here is a sample conversation about "bookcase" between the GPT-4 agent and the simulated shopper in the ChatShop task.

AGENT: Do you have a preference for a specific type of cabinet such as wine, file, or general storage?

SHOPPER: General storage cabinet.

AGENT: Do you prefer a specific color for your general storage cabinet?

SHOPPER: White preferred.

AGENT: Do you prefer a cabinet that is freestanding or one that mounts on the wall?

SHOPPER: Freestanding.

AGENT: Do you need the cabinet to have a specific height or size to fit in your space?

SHOPPER: Standard size is fine.

AGENT: Would you like a cabinet with doors, drawers, or both for your storage needs?

SHOPPER: Doors only.

Another example regarding "hair chalk".

AGENT: Do you have any preference for the number of colors in the hair chalk set? Or are there any specific colors you're looking for?

SHOPPER: 6 colors.

AGENT: Do you prefer the hair chalk to be easily washable or are you looking for something that might last a bit longer before washing out?

SHOPPER: Washable hair chalk.

AGENT: Are you looking for hair chalk that comes with additional hair accessories or are you only interested in the hair chalk itself?

SHOPPER: Hair chalk only.

AGENT: Do you prefer hair chalk that is specifically marketed for kids and teens, or does it not matter to you as long as the product meets your requirements?

SHOPPER: Doesn't matter.

AGENT: Is there a specific brand or any additional attribute you are looking for in the hair chalk set?

SHOPPER: Non-toxic and easy clean.

Strategy	**Open-ended**		**Instance**	
CoT	w/o	w/	w/o	w/
no q	43.4	45.6	43.4	45.6
auto q	40.6	62.7	40.4	51.6
all q	63.7	61.3	48.3	47.1
interleave	64.3	**68.2**	51.1	51.3

Table 5: Avg. rewards of the GPT-3.5 based agents with different interaction strategies and both open-ended communication channels. *no q* is the non-interactive baselines.

B.5 Human Evaluation

We recruit 8 participants (university students) to play the role of the shopper in the human study. Each participant is asked to complete half to one hour of the ChatShop task. The participants are compensated on average $12 for their time.

C Additional Results

We show additional results of the GPT-3.5 based agents with the instance-based communication channel in Table 5.

C.1 Error Types Classification

For classification of error types, we use the GPT-4 model to tag failed trajectories with the likely causes of failure as a multi-label classification problem. We design a prompt consists of the flattened conversation history, the agent selected product, the goal product, and fine-grained rewards (i.e., title similarity, attribute/option coverage seperately). The GPT-4 model judges the relevance of each error type based on the textual description of them and the episode context.[9]

We define the five error types as follows.

1. **Reversion**: the agent loses track of shopper specified requirements. In the context of LLM agents, this is often caused by the agent's inability to robustly recall information across long contexts (Liu et al., 2023a).

2. **Misinterpretation**: the agent fails to understand the shopper mentioned specification. As a realistic shopping scenario, our task covers a diverse range of products and attributes and grounded understanding of the shopper's intention can be challenging and error-prone.

3. **Insufficient information gathering**: the agent does not gather enough information to

[9] We use the `gpt-4-0125-preview` version.

locate the correct product, causing important attributes/options to be missing. This error is associated with the agent's lack of strategic information seeking and overconfidence in making decisions based on partial information.

4. **Repeated questions or search**: the agent asks the same question or searches the same query repeatedly, leading to inefficient actions. Language models are known to have a tendency to repeat themselves in long context (Holtzman et al., 2020).

5. **Misleading user**: the shopper makes mistakes, being inconsistent or unclear. As a dynamic and interactive environment, it is natural that the shopper makes mistakes or corrects themselves. The agent should be able to tolerate certain level of noise and handle these cases gracefully. This also serves as a sanity check for the simulated shopper.

D ChatShop Prompts

D.1 Shopper Prompt

System Prompt:

```
You are playing the role of a shopper. While interacting, avoid explicitly stating the name of the product you intend to purchase. However, if prompted for specific related information, you may provide descriptions using alternative expressions and indirect references.

Product name: OWYN - 100% Vegan Plant-Based Protein Shakes | Cold Brew Coffee, 12 Fl Oz | Dairy-Free, Gluten-Free, Soy-Free, Tree Nut-Free, Egg-Free, Allergy-Free, Vegetarian

Important attributes: gluten free
```

Sample User Prompt:

```
Do you have any allergies?
```

D.2 Agent Prompt

System Prompt:

```
Your role is to guide users through an online shopping experience, helping them find products that best fit their needs. When a user specifies certain attributes, you analyze these to sift through the available products, based on detailed product descriptions. There are three key actions:

- `search[query]`: At the start, and whenever necessary, you can initiate a search using the website's BM25 search engine. Price can't be searched. This search yields a list of products, each with a unique description and index number. You may perform this action multiple times to refine the search based on evolving user requirements.

- `select[item_index]`: When the user's criteria clearly match a single product, you finalize your response with `select[]`. Here, `item_index` refers to the unique number of the identified product.

- `question[question_content]`: When multiple products fit the user's described attributes, or when more information is needed for a precise decision, you narrow down the choices with `candidates[0, 1, 2]` for example, listing the indexes of potential matches. Concurrently, you should pose questions to the user for further clarification.
```

Sample User Prompt:

```
Goal: i need to find a small end table that is easy to assemble; pick a blue-coated steel frame that won't rust\nThe next action is
```

D.3 Subject Extraction Prompt

System Prompt:

```
You assist users in extracting the main target from their search queries by removing all product attributes. Your response only contain the cleaned query.
```

Sample User Prompt:

```
User Query: "i want a noise cancelling cosycost usb microphone"
```

Sample Assistant Prompt:

```
microphone
```

D.4 Attribute Removal Prompt

System Prompt:

```
You assist users in refining their
    search queries by removing
    specific product attributes. When
    a user provides a query, you must
    identify and remove any attribute
    mentioned that is listed in the
    provided attribute removal list.
    The cleaned query should still be
    fluent. Your response only contain
    the cleaned query.
```

Sample User Prompt:

```
User Query: "i want a noise cancelling
    cosycost usb
    microphone"\nAttribute Removal
    List: [noise cancelling]
```

Sample Assistant Prompt:

```
i want a noise cancelling cosycost usb
    microphone
```