ProductAgent: Benchmarking Conversational Product Search Agent with Asking Clarification Questions

Jingheng Ye[1], **Yong Jiang**[2*], **Xiaobin Wang**[2], **Yinghui Li**[1],
Yangning Li[1,3], **Hai-Tao Zheng**[1,3*], **Pengjun Xie**[2], **Fei Huang**[2]
[1]Tsinghua Shenzhen International Graduate School, Tsinghua University,
[2]DAMO Academy, Alibaba Group, [3]Peng Cheng Laboratory
yejh22@mails.tsinghua.edu.cn
{yongjiang.jy,xuanjie.wxb}@alibaba-inc.com

Abstract

Online shoppers often initiate their journey with only a vague idea of what they need, forcing them to iterate over search results until they eventually discover a suitable product. We formulate this scenario as *product demand clarification*: starting from an ambiguous query, an agent must iteratively ask clarifying questions, progressively refine the user's intent, and retrieve increasingly relevant items. To tackle this challenge, we present **ProductAgent**, a fully autonomous conversational information-seeking agent that couples large language models with a set of domain-specific tools. ProductAgent maintains a structured memory of the dialogue, summarizes candidate products into concise feature statistics, generates strategic clarification questions, and performs retrieval over hybrid (symbolic + dense) indices in a closed decision loop. To measure real-world effectiveness, we further introduce **PROCLARE**, a PROduct CLArifying REtrieval benchmark that pairs ProductAgent with an LLM-driven user simulator, thereby enabling large–scale and reproducible evaluation without human annotation. On 2,000 automatically generated sessions, retrieval metrics improve monotonically with the number of turns, validating that ProductAgent captures and refines user intent through dialogue.

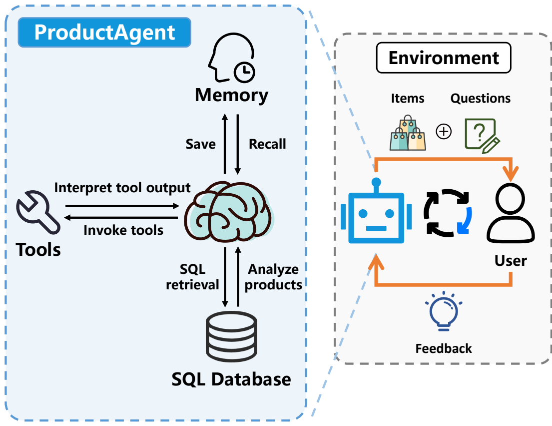

Figure 1: Simplified illustration of our ProductAgent.

1 Introduction

Intelligent agents are traditionally developed under the convenient assumption that users articulate their needs unambiguously (Li et al., 2025a; Ye et al., 2025c; Chu et al., 2025). Reality is very different: when people face an unfamiliar domain, they often express only partial, underspecified preferences and refine them through interaction (Vats et al., 2024; Yi et al., 2024). E-commerce is a prime example. Unlike in physical stores—where a salesperson can probe, explain, and recommend—online shoppers must repeatedly reformulate queries, scroll through pages of results, and compare heterogeneous product facets such as brand, colour, or style (Yang et al., 2018). The absence of interactive clarification, therefore, translates into lost time and, ultimately, lost revenue.

Recent progress in *conversational information seeking* (CIS) suggests a remedy: agents that elicit missing constraints via natural language questions and update recommendations on the fly (Zhang et al., 2018; Rahmani et al., 2023). Yet building such agents for product search remains difficult. An effective solution must (1) access large-scale catalogues, (2) understand nuanced user utterances, (3) decide which product facets to clarify next, and (4) retrieve items under tight latency constraints—all while the user's goals evolve.

We address these challenges through **ProductAgent**, a virtual shopping assistant whose architecture is illustrated in Figure 1. ProductAgent operates an autonomous loop that intertwines three core components. (1) *Product databases* store catalogue entries in both structured and vector formats, enabling hybrid symbolic-dense retrieval. (2) A *memory module* tracks the dialogue context-clarification

*Corresponding authors: Yong Jiang and Hai-Tao Zheng. (E-mail: yongjiang.jy@alibaba-inc.com, zheng.haitao@sz.tsinghua.edu.cn)

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 383–398
November 4-9, 2025 ©2025 Association for Computational Linguistics

questions, answers, and free-form chat, so that decisions remain sensitive to the entire interaction history. (3) A set of *tools* exposes retrieval, summarization, and question-generation primitives. Carefully engineered prompts orchestrate these tools with large language models (LLMs), allowing the agent to reason about which facet to clarify, pose a succinct question, and fetch candidate items that satisfy the newly specified constraints (Yan et al., 2025).

To evaluate our ProductAgent, we propose the **PROduct CLArifying REtrieval (PROCLARE)** benchmark, inspired by recent progress in automatic task-oriented evaluation (Zhou et al., 2023; Semnani et al., 2023). A GPT-based user simulator (Terragni et al., 2023; Sekulić et al., 2024) is conditioned on a hidden target item and replies to the agent's questions exactly as a consumer would. This setup produces realistic and large-scale conversation logs without manual effort, enabling rigorous measurement of retrieval quality as the dialogue unfolds. Experiments on 2,000 simulated sessions show that ProductAgent consistently narrows the candidate set and boosts retrieval accuracy with every additional turn.

In summary, our contributions are threefold:

(1) We propose ProductAgent, an end-to-end conversational agent that clarifies product demand by questioning users and retrieving items.

(2) We introduce PROCLARE, a conversational retrieval-focused benchmark to ensure a reliable and automatic evaluation pipeline for the product demand clarification task.

(3) Extensive experiments demonstrate that clarification questions materially improve retrieval effectiveness.

2 Related Work

Task-oriented dialogue systems aim to fulfill concrete user goals, yet they often receive underspecified or ambiguous requests because users may lack the domain knowledge to express precise constraints (Xu et al., 2024; Li et al., 2024c; Tang et al., 2025; Li et al., 2025b). While early work attempted to address this mismatch implicitly, for instance by returning diversified results, recent research advocates for *explicit* clarification as a more reliable strategy. Classical information-retrieval studies generate follow-up questions to disambiguate web queries (Zamani et al., 2020; Kuhn et al., 2022). Extending this idea to personal assistants, MAS2S (Feng et al., 2023) conditions question generation on a user profile and a task graph, while CAMBIGNQ (Lee et al., 2023) introduces a benchmark of naturally ambiguous questions alongside a pipeline designed to solicit missing facets. The advent of large language models has amplified interest in clarification (Ye et al., 2025b,a). Prompt-based or fine-tuned LLMs have demonstrated the ability to ask high-utility follow-up questions in search (Wu, 2024; Hu et al., 2024), reading comprehension (Erbacher and Soulier, 2023; Li et al., 2024b; Zou et al., 2025; Li et al., 2023a), and multi-hop reasoning (Zhang et al., 2024; Huang et al., 2023b). In the commerce domain, Vedula et al. (2024) leverage LLMs to propose product-specific clarifiers, whereas the concurrent *SalesAgent* (Chang and Chen, 2024) fine-tunes an LLM to emulate salesperson tactics. Despite their versatility, these approaches rely almost exclusively on the model's internal knowledge; consequently, their questions may be inaccurate, outdated, or misaligned with the live catalogue.

Our work departs from this paradigm by coupling the reasoning capabilities of an LLM with external artifacts: a hybrid product index, a dialogue memory, and specialized retrieval tools. Offloading factual lookups to dedicated databases reduces the inferential burden on the LLM and yields clarification questions that are both precise and grounded in the current inventory, ultimately leading to higher downstream retrieval accuracy.

Due to the intricacies of real-world scenarios (Su et al., 2025; Qin et al., 2025; Yu et al., 2024a), several benchmarks recommend the inclusion of user simulators to enhance the reliability and efficiency of evaluation for agents or task-oriented dialogue systems. For instance, WebArena (Zhou et al., 2023) generates unique user profiles to emulate real-world scenarios where users often have disparate experiences. WikiChat (Semnani et al., 2023) leverages LLMs as simulated users to synthesize dialogue flows for cost-effectiveness and superior quality. The authors generate a simulated 10-turn dialogue on which they conduct a comprehensive evaluation of the developed chatbot.

3 Product Demand Clarification Task

The product demand clarification task models the typical situation in which an online customer has selected a coarse product category yet is still unable

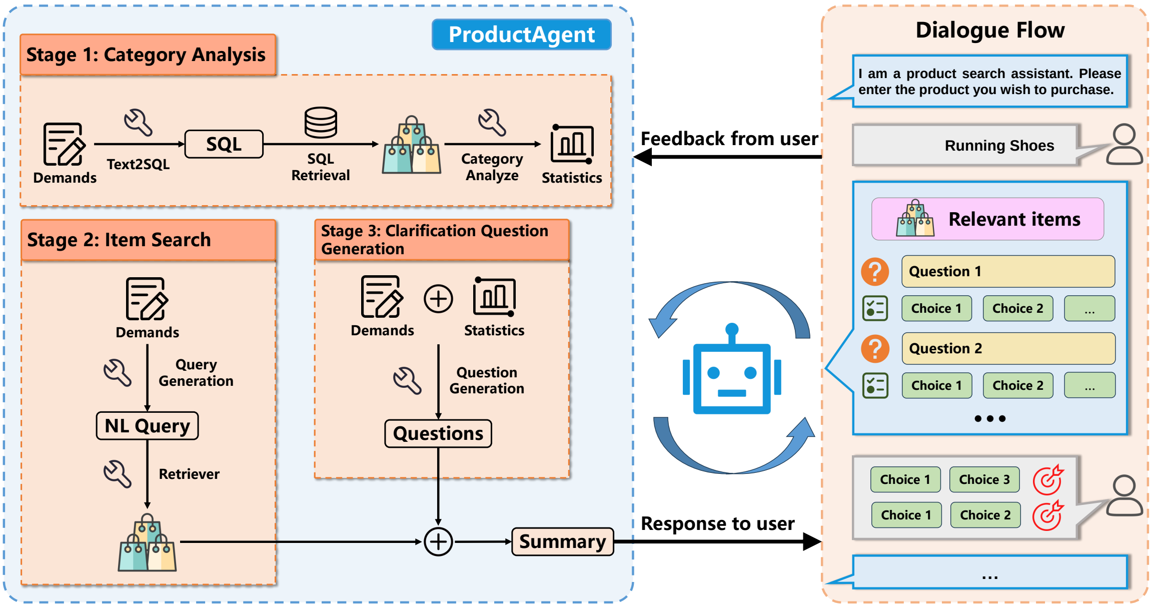

Figure 2: Overview of our proposed ProductAgent.

to articulate concrete purchasing requirements such as brand, size, style, or price range. Bridging this information gap is crucial: if the system can elicit missing constraints quickly, both user satisfaction and conversion rates increase. We therefore cast the interaction as a mixed-initiative conversation in which an intelligent agent iteratively asks clarification questions, assimilates the user's feedback, and retrieves an updated set of candidate items.

The task is structured as a conversation, where an agent actively engages with a user. Formally, given a product category, the conversation between them is defined as follows:

$$D = \{U_1, A_1, P_1, U_2, A_2, P_2, \cdots\}, \qquad (1)$$

where at turn t the user utterance U_t expresses her current preferences, the agent utterance A_t contains the follow-up questions, and P_t is the set of products returned after integrating all information known so far. In our implementation, each A_t comprises $n = 3$ multiple-choice clarification questions $\{Q_{t1}, Q_{t2}, Q_{t3}\}$, each accompanied by answer candidates that serve both as disambiguation aids and as concrete examples. Immediately after posting its questions, the agent also presents P_t to provide instant feedback; the user may answer by selecting one of the suggested options or by typing a free-form response if none of them suffice.

This seemingly simple interaction masks several intertwined challenges for building shopping assistants: (1) Generating non-trivial and informative questions demands extensive product knowledge and an ability to reason over long-tail attributes. (2) The agent must interpret possibly noisy answers and translate them into structured constraints that guide retrieval under tight latency budgets. (3) Progress in the field is hindered by the absence of an automated and reproducible evaluation protocol that quantifies how well an agent converges on the user's true intent. The following sections tackle the first two challenges by introducing ProductAgent (Section 4) and address the third by proposing a large-scale simulation benchmark (Section 5).

4 ProductAgent

4.1 Overview of ProductAgent

This section outlines the proposed solution ProductAgent for the given task. ProductAgent carries out conversational loops broken down into three primary steps, as illustrated in Figure 2. Before starting the loops, the user initiates the conversation by entering a product category as a simple query. We will provide an overview of ProductAgent in this section and detail our implementation in Section 4.3.

4.2 Autonomous Dialogue Loop

Figure 2 depicts the closed control loop that enables ProductAgent to refine a user's purchasing goal over successive turns. Each session starts with a terse natural-language query that specifies only a coarse product category (e.g., "Running shoes"). Thereafter, the agent repeatedly executes three phases, continuously injecting the freshly acquired knowledge back into the loop.

(i) Category Analysis. At the beginning of every turn, ProductAgent assembles a structured representation of *all* constraints expressed so far, draws a hybrid symbolic-dense query, and retrieves a small pool of catalog entries that plausibly satisfy the current demand. The pool is not yet ranked for final recommendation; instead, it is processed by the CATEGORYANALYZING tool, which converts raw product facets (brand, material, ...) into compact statistics. These statistics act as a transient knowledge base that grounds the subsequent reasoning steps and prevents the LLM from hallucinating attributes that do not appear in the catalog.

(ii) Item Search. Conditioned on the up-to-date demand profile collected from dialogue, the agent invokes the QUERYGENERATION tool to transform the structured constraints into a concise natural-language query. The query is forwarded to the retriever, which ranks the full product catalog and returns the k most relevant items. The resulting set P_t is immediately shown to the user, providing transparency and instant feedback.

(iii) Clarification Question Generation. Finally, the QUESTIONGENERATION tool leverages both the statistics from step (i) and the evidence retrieved in step (ii) to craft n multiple-choice clarification questions. Answer candidates are drawn directly from the catalog, ensuring that every option can be satisfied by at least one real item. The user either selects one of the provided choices or writes a free-form answer, thereby enriching the demand profile that seeds the next loop iteration. Thanks to this mixed-initiative design, the conversation gradually moves from vague intent to precise, executable constraints, and retrieval accuracy increases monotonically with the number of turns (See Section 6.2).

4.3 System Components

The above loop is realized through three tightly coupled components.

Databases. Each catalog entry is stored twice: (1) in an SQL database that supports exact-match filters over structured facets, and (2) in a vector database that indexes sentence-level embeddings of the textual description. ProductAgent therefore chooses the substrate that best fits the current subtask: SQL for high-precision constraint satisfaction in step (i) and dense retrieval for semantic ranking in step (ii).

Tools. Table 1 summarizes the five specialized tools that extend the LLM's capabilities. The TEXT2SQL tool converts the remembered constraints into executable SQL, whereas CATEGORYANALYZING distills the retrieved pool into statistics that drive question generation. QUERYGENERATION produces dense-retrieval queries, and QUESTIONGENERATION formulates clarification questions. Finally, RETRIEVER retrieves relevant product items based on the latest demand profile.

Memory. All questions, answers, and issued retrieval queries are stored as structured objects in a memory buffer. When constructing prompts for any tool, ProductAgent injects only the fields that are relevant to the tool's function, thereby keeping the context window short and inference latency low. The memory is continually updated as the dialogue progresses, ensuring the intelligent agent can effectively respond to dynamically changing user requirements. This design is crucial as it enables the agent to recall user preferences and provide personalized assistance.

5 The PROCLARE Benchmark

Although recent IR benchmarks have significantly advanced static search, their single-turn setting prevents a rigorous assessment of conversational agents, whose strength lies in iteratively narrowing the search space through interaction. PROCLARE is designed to fill this gap: it supplies a large, realistic product corpus, a conversation–oriented evaluation protocol, and an automatic metric suite that jointly quantify an agent's ability to clarify user intent and retrieve the correct item over multiple turns.

5.1 Document Set

We first construct a million–scale product corpus that serves as the knowledge base for all experiments. Starting from AliMe KG (Li et al., 2020), an e-commerce knowledge graph maintained by Alibaba, we randomly sample 50,000 items for each of 20 product categories, yielding 1,000,000 documents in total. Each item is pre-processed with a Named Entity Recognition (NER) pipeline (Wang et al., 2021) to normalize attribute names across categories. The resulting tuples are stored both in

Tool	Description	Input	Output
Text2SQL	Generate SQL query	Demands	SQL Query
Category Analyze	Summarize a certain category	Product items	Category statistics
Query Generation	Generate NL query	Demands	NL query
Retriever	Retrieve items relevant to demands	NL query	Product items
Question Generation	Generate clarification questions	Demands + Category statistics	Clarification questions

Table 1: Available tools of ProductAgent.

Setting	Documents	Query	Len. of Query
Trad.	1,000,000	2,000	27.02
Conv.	1,000,000	10,000	8.59 / 16.45 / 27.40 / 37.33 / 45.03

Table 2: Statistics of the PROCLARE benchmark. We report the average length of queries generated by GPT-4.

a relational SQL database and in a vector store, which enables efficient hybrid retrieval. Dataset statistics and full feature lists are provided in Table 2 and Appendix A.1.

5.2 Retrieval Strategies

To isolate the contribution of the conversational layer from that of the underlying retrieval model, we support three representative retrievers: the term–based BM25 (Robertson et al., 2009), the dense encoder GTE[1] and CoROM (Long et al., 2022). All retrievers can optionally employ a lightweight BGE reranker and a reciprocal-rank fusion scheme to boost final rankings. Implementation details appear in Appendix A.2.

5.3 Conversational Evaluation Protocol

In PROCLARE, each dialogue starts with the user simulator stating the desired product category, mimicking the uncertainty of real shoppers. The agent then alternates with the user for five clarification turns, asking questions, retrieving candidates, and refining its belief state. We build the simulator on `gpt-3.5-turbo-0125` (Terragni et al., 2023; Sekulić et al., 2024) that is conditioned on the ground-truth item yet constrained to answer solely with information prompted by the agent, preventing leakage of hidden attributes. This design produces 2,000 automatically generated dialogues, covering the full product spectrum while eliminating the prohibitive cost of human annotation. Prompts of ProductAgent and user simulator, as well as dialogue snippets, are provided in Appendix B.

[1] `https://www.alibabacloud.com/help/en/model-studio/developer-reference/model-introduction-6`

LLM	Retriever	Retrieve		Rerank
		HIT@10	MRR@10	MRR@10
GPT-3.5	BM25	**35.04**	**27.26**	**25.69**
	GTE	8.49	4.95	7.18
	CoROM	12.48	7.96	10.79
	B + G	32.35	16.12	25.21
	B + C	32.54	17.66	25.27
	G + C	7.61	4.21	6.67
GPT-4	BM25	**39.48**	**32.00**	**30.20**
	GTE	8.27	4.92	7.17
	CoROM	13.86	9.11	12.54
	B + G	36.93	18.36	29.91
	B + C	37.02	20.37	30.13
	G + C	7.57	4.15	6.80
Qwen	BM25	**31.58**	**25.24**	**24.85**
	GTE	16.45	10.56	13.82
	CoROM	20.71	13.80	17.77
	B + G	30.65	16.78	24.61
	B + C	30.79	17.98	24.79
	G + C	14.26	7.74	12.36

Table 3: Retrieval performance of the conversational setting. We report results of fusion retrievers represented as X + Y, such as B + G (BM25 + GTE).

5.4 Automatic Metrics

Following standard IR practice, we measure retrieval quality with Mean Reciprocal Rank at ten (MRR@10) and Hit Rate at ten (HIT@10). Because reranking affects only the order, not the membership, of the top-k set, we report MRR@10 for reranked outputs and reuse the original HIT@10. Crucially, we record these metrics after each turn, enabling a fine-grained analysis of how clarification questions translate into incremental search gains.

6 Experiments

6.1 Main Results

Three distinct variants of ProductAgent are instantiated by plugging in different LLM backbones, namely `Qwen-max-0107`, `gpt-3.5-turbo-0125`, and `gpt-4-turbo-0409`. Each variant functions in a fully autonomous manner, invoking the tool chain described in Section 4.3. The quality of retrieval is assessed after each dialogue turn and subsequently averaged across 2,000 simulated sessions, each con-

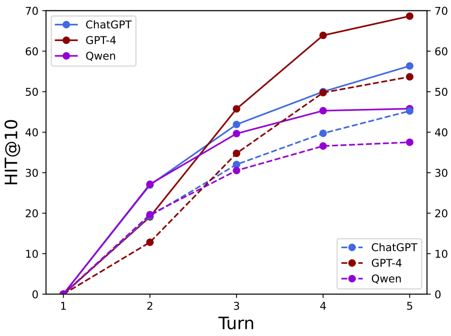

Figure 3: Retrieval performance of ProductAgent driven by different LLMs with increasing turns using BM25.

taining five dialogue turns.

Table 3 shows that the classical BM25 retriever consistently outperforms the two dense retrievers under all LLM backbones. Because the query-generation tool transforms user feedback into short option-like phrases that often overlap with product attributes, exact-match signals become unusually reliable and therefore favour BM25. When a dense retriever is employed, reranking the top-k candidates with an LLM-based ranker markedly improves MRR@10. In contrast, BM25 already produces a well-ordered list and gains little from an additional reranking stage. Combining symbolic and dense scores through retriever fusion does not yield further gains, suggesting that the complementary information captured by the two families of models is still limited in this domain.

6.2 Effect of Interaction Turns

Retrieval effectiveness grows monotonically with the number of dialogue turns, as illustrated in Figure 3. The first turn cannot retrieve any ground-truth items because only the coarse product category is known. Once the agent starts asking clarification questions, Recall@10 and MRR@10 rise steadily, confirming that the additional constraints extracted from the conversation are successfully converted into more discriminative search queries. Interestingly, the `gpt-4-turbo` backbone lags slightly behind the others after the second turn but surpasses them from the third turn onwards; a qualitative inspection suggests that `gpt-4-turbo` prefers broader questions early on, postponing fine-grained constraints to later turns.

6.3 Ablation Study

Because the question-generation tool relies on dynamic product statistics, we examine four ablated

Retriever	HIT@10	MRR@10
w/o Statistics	15.60	10.69
Random	39.50	19.54
BM25	**47.00**	**38.51**
CoROM	45.00	38.09
Text2SQL	39.90	32.40

Table 4: Ablation results of different strategies of acquiring statistics.

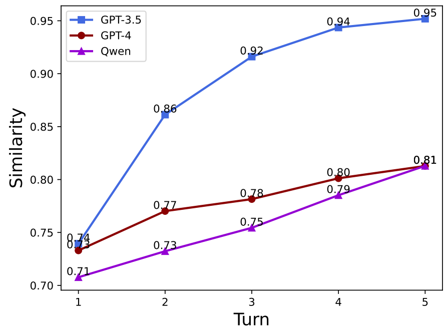

Figure 4: Similarity of synthesized clarification questions computed by BERTScore.

versions of ProductAgent: (1) no statistics, (2) statistics from randomly sampled items, (3) statistics obtained with BM25 instead of Text2SQL, and (4) statistics obtained with CoROM. Results in Table 4 indicate that accurate, demand-aware statistics are crucial. Removing statistics or ignoring current user constraints degrades MRR@10 by more than three points, whereas collecting statistics via BM25 or CoROM mitigates the loss and even surpasses the original Text2SQL variant.

6.4 Similarity of Synthesized Clarification Questions

We observe that ProductAgent sometimes generates clarification questions that overlap with those previously presented, which may negatively impact the task effectiveness and overall user satisfaction. Hence, we qualitatively measure the similarity among clarification questions proposed in different dialogue turns by using BERTScore (Zhang* et al., 2020). BERTScore computes a similarity score for each token in the candidate sentence by comparing it to each token in the reference sentence. In the experiment, we concatenate the question content and its answer choices into a sentence that will be input into BERTScore. By treating all other questions and choices as references, we compute the similarity score for each synthesized clarification question against others.

As Figure 4 illustrates, we discern an upward trend in the similarity of synthesized clarification questions as dialogues progress. This suggests that although retrieval performance continues to enhance over ten dialogue turns, we may still be able to enhance performance by boosting the diversity of clarification questions. Additionally, we observe that the agent powered by GPT-3.5 is more inclined to fabricate similar questions than those powered by GPT-4 and Qwen-max, even though the former achieves higher retrieval scores than the agent powered by Qwen-max when both utilize BM25 for item search.

7 Conclusion

This paper formulates the task of product demand clarification and presents the solution called ProductAgent. The agent poses strategic clarification questions to pinpoint user demands. To implement the evaluation automatically and quantitatively, we introduce the PROCLARE benchmark with the aid of a user simulator acted by an LLM. Experiments indicate that ProductAgent significantly enhances the retrieval performance with increasing dialogue turns, regardless of applied retrievers.

Limitations

Absence of human users. In this paper, we employ the user simulation technique to facilitate a more consistent and standardized evaluation process. Utilizing this approach allows for all the experiments conducted within our research to be completely void of any human involvement, contributing to cost efficiency and saving time. We also acknowledge that potentially unidentified biases may arise from the hallucination effect of LLM-based user simulators within our evaluation pipeline.

Limitations of datasets. Our dataset was derived from the AliMe KG, encompassing multiple dimensions of products. Nonetheless, it does not provide comprehensive information that could be crucial to customers' purchase decisions, such as product pricing and customer reviews. Future research could delve into more diverse and higher-quality datasets to overcome these deficiencies.

Controllability of clarification questions. We encourage ProductAgent to ask clarification questions following in-context learning augmented with summarized statistics, without explicit and strong constraints. This strategy is not always optimal for all situations. In future work, we will explore more effective planning strategies that can handle the task from a global perspective, providing a pathway for potential enhancements to create more intelligent conversational agents.

Lack of analysis of prompt sensitivity. The impact of prompt design on the performance of LLMs, often demonstrated as prompt sensitivity, is widely recognized. In our research, however, we did not conduct a systematic examination concerning the sensitivity of prompts used to develop the agent and user simulator. Detailed engineering of prompts could serve as a valuable measure for enhancing the performance of the ProductAgent in future studies.

Ethics Statement

We are aware that our proposed ProductAgent may be potentially misused for improper purposes, such as privacy data collection and excessive propaganda. However, this vulnerability is not unique to our approach but is a common threat to many LLM-based applications. It also highlights the significance of conducting appropriate regulations and enhancing the safety of LLMs in the future.

Acknowledgements

This research is supported by National Natural Science Foundation of China (Grant No.62276154); Research Center for Computer Network (Shenzhen) Ministry of Education, the Natural Science Foundation of Guangdong Province (Grant No.2023A1515012914 and 440300241033100801770); Basic Research Fund of Shenzhen City (Grant No.JCYJ20210324120012033, JCYJ20240813112009013 and GJHZ20240218113603006); The Major Key Project of PCL for Experiments and Applications (PCL2024A08).

References

Wen-Yu Chang and Yun-Nung Chen. 2024. Injecting salesperson's dialogue strategies in large language models with chain-of-thought reasoning. *arXiv preprint arXiv:2404.18564*.

Zhendong Chu, Shen Wang, Jian Xie, Tinghui Zhu, Yibo Yan, Jingheng Ye, Aoxiao Zhong, Xuming Hu, Jing Liang, Philip S Yu, et al. 2025. Llm agents for education: Advances and applications. *arXiv preprint arXiv:2503.11733*.

Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 758–759.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Pierre Erbacher and Laure Soulier. 2023. Circle: Multi-turn query clarifications with reinforcement learning. *arXiv preprint arXiv:2311.02737*.

Yue Feng, Hossein A Rahmani, Aldo Lipani, and Emine Yilmaz. 2023. Towards asking clarification questions for information seeking on task-oriented dialogues. *arXiv preprint arXiv:2305.13690*.

Xuming Hu, Xiaochuan Li, Junzhe Chen, Yinghui Li, Yangning Li, Xiaoguang Li, Yasheng Wang, Qun Liu, Lijie Wen, Philip S Yu, et al. 2024. Evaluating robustness of generative search engine on adversarial factual questions. *arXiv preprint arXiv:2403.12077*.

Haojing Huang, Jingheng Ye, Qingyu Zhou, Yinghui Li, Yangning Li, Feng Zhou, and Hai-Tao Zheng. 2023a. A frustratingly easy plug-and-play detection-and-reasoning module for Chinese spelling check. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Singapore.

Shulin Huang, Shirong Ma, Yinghui Li, Mengzuo Huang, Wuhe Zou, Weidong Zhang, and Hai-Tao Zheng. 2023b. Lateval: An interactive llms evaluation benchmark with incomplete information from lateral thinking puzzles. *arXiv preprint arXiv:2308.10855*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2022. Clam: Selective clarification for ambiguous questions with generative language models. *arXiv preprint arXiv:2212.07769*.

Dongryeol Lee, Segwang Kim, Minwoo Lee, Hwanhee Lee, Joonsuk Park, Sang-Woo Lee, and Kyomin Jung. 2023. Asking clarification questions to handle ambiguity in open-domain QA. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11526–11544, Singapore. Association for Computational Linguistics.

Feng-Lin Li, Hehong Chen, Guohai Xu, Tian Qiu, Feng Ji, Ji Zhang, and Haiqing Chen. 2020. Alimekg: Domain knowledge graph construction and application in e-commerce. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2581–2588.

Yangning Li, Tingwei Lu, Hai-Tao Zheng, Yinghui Li, Shulin Huang, Tianyu Yu, Jun Yuan, and Rui Zhang. 2024a. Mesed: A multi-modal entity set expansion dataset with fine-grained semantic classes and hard negative entities. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 8697–8706.

Yangning Li, Weizhi Zhang, Yuyao Yang, Wei-Chieh Huang, Yaozu Wu, Junyu Luo, Yuanchen Bei, Henry Peng Zou, Xiao Luo, Yusheng Zhao, et al. 2025a. Towards agentic rag with deep reasoning: A survey of rag-reasoning systems in llms. *arXiv preprint arXiv:2507.09477*.

Yinghui Li, Haojing Huang, Jiayi Kuang, Yangning Li, Shu-Yu Guo, Chao Qu, Xiaoyu Tan, Hai-Tao Zheng, Ying Shen, and Philip S Yu. 2025b. Refine knowledge of large language models via adaptive contrastive learning. *arXiv preprint arXiv:2502.07184*.

Yinghui Li, Haojing Huang, Shirong Ma, Yong Jiang, Yangning Li, Feng Zhou, Hai-Tao Zheng, and Qingyu Zhou. 2023a. On the (in) effectiveness of large language models for chinese text correction. *arXiv preprint arXiv:2307.09007*.

Yinghui Li, Yangning Li, Yuxin He, Tianyu Yu, Ying Shen, and Hai-Tao Zheng. 2022. Contrastive learning with hard negative entities for entity set expansion. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1077–1086.

Yinghui Li, Shang Qin, Jingheng Ye, Shirong Ma, Yangning Li, Libo Qin, Xuming Hu, Wenhao Jiang, Hai-Tao Zheng, and Philip S Yu. 2024b. Rethinking the roles of large language models in chinese grammatical error correction. *arXiv preprint arXiv:2402.11420*.

Yinghui Li, Qingyu Zhou, Yuanzhen Luo, Shirong Ma, Yangning Li, Hai-Tao Zheng, Xuming Hu, and Philip S Yu. 2024c. When llms meet cunning texts: A fallacy understanding benchmark for large language models. *Advances in Neural Information Processing Systems*, 37:112433–112458.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023b. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.

Dingkun Long, Qiong Gao, Kuan Zou, Guangwei Xu, Pengjun Xie, Ruijie Guo, Jian Xu, Guanjun Jiang, Luxi Xing, and Ping Yang. 2022. Multi-cpr: A multi domain chinese dataset for passage retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3046–3056.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.

Shang Qin, Jingheng Ye, Yinghui Li, Hai-Tao Zheng, Qi Li, Jinxiao Shan, Zhixing Li, and Hong-Gee Kim. 2025. Cl ^{2} gec: A multi-discipline benchmark for continual learning in chinese literature grammatical error correction. *arXiv preprint arXiv:2509.13672*.

Hossein A. Rahmani, Xi Wang, Yue Feng, Qiang Zhang, Emine Yilmaz, and Aldo Lipani. 2023. A survey on asking clarification questions datasets in conversational systems. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2698–2716, Toronto, Canada. Association for Computational Linguistics.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Ivan Sekulić, Silvia Terragni, Victor Guimarães, Nghia Khau, Bruna Guedes, Modestas Filipavicius, André Ferreira Manso, and Roland Mathis. 2024. Reliable llm-based user simulator for task-oriented dialogue systems. *arXiv preprint arXiv:2402.13374*.

Sina Semnani, Violet Yao, Heidi Chenyu Zhang, and Monica Lam. 2023. Wikichat: Stopping the hallucination of large language model chatbots by few-shot grounding on wikipedia. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Jiamin Su, Yibo Yan, Fangteng Fu, Han Zhang, Jingheng Ye, Xiang Liu, Jiahao Huo, Huiyu Zhou, and Xuming Hu. 2025. Essayjudge: A multi-granular benchmark for assessing automated essay scoring capabilities of multimodal large language models. *arXiv preprint arXiv:2502.11916*.

Zeqi Tan, Shen Huang, Zixia Jia, Jiong Cai, Yinghui Li, Weiming Lu, Yueting Zhuang, Kewei Tu, Pengjun Xie, Fei Huang, et al. 2023. Damo-nlp at semeval-2023 task 2: A unified retrieval-augmented system for multilingual named entity recognition. *arXiv preprint arXiv:2305.03688*.

Jiwei Tang, Zhicheng Zhang, Shunlong Wu, Jingheng Ye, Lichen Bai, Zitai Wang, Tingwei Lu, Jiaqi Chen, Lin Hai, Hai-Tao Zheng, et al. 2025. Gmsa: Enhancing context compression via group merging and layer semantic alignment. *arXiv preprint arXiv:2505.12215*.

Silvia Terragni, Modestas Filipavicius, Nghia Khau, Bruna Guedes, André Manso, and Roland Mathis. 2023. In-context learning user simulators for task-oriented dialog systems. *arXiv preprint arXiv:2306.00774*.

Vanshika Vats, Marzia Binta Nizam, Minghao Liu, Ziyuan Wang, Richard Ho, Mohnish Sai Prasad, Vincent Titterton, Sai Venkat Malreddy, Riya Aggarwal, Yanwen Xu, et al. 2024. A survey on human-ai teaming with large pre-trained models. *arXiv preprint arXiv:2403.04931*.

Nikhita Vedula, Oleg Rokhlenko, and Shervin Malmasi. 2024. Question suggestion for conversational shopping assistants using product metadata. *arXiv preprint arXiv:2405.01738*.

Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. 2021. Improving named entity recognition by external context retrieving and cooperative learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1800–1812, Online. Association for Computational Linguistics.

Jie JW Wu. 2024. Large language models should ask clarifying questions to increase confidence in generated code.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding.

Zhikun Xu, Yinghui Li, Ruixue Ding, Xinyu Wang, Boli Chen, Yong Jiang, Hai-Tao Zheng, Wenlian Lu, Pengjun Xie, and Fei Huang. 2024. Let llms take on the latest challenges! a chinese dynamic question answering benchmark. *arXiv preprint arXiv:2402.19248*.

Yibo Yan, Shen Wang, Jiahao Huo, Jingheng Ye, Zhendong Chu, Xuming Hu, Philip S Yu, Carla Gomes, Bart Selman, and Qingsong Wen. 2025. Position: Multimodal large language models can significantly advance scientific reasoning. *arXiv preprint arXiv:2502.02871*.

Yunlun Yang, Yu Gong, and Xi Chen. 2018. Query tracking for e-commerce conversational search: A machine comprehension perspective. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1755–1758.

Jingheng Ye, Yinghui Li, Yangning Li, and Hai-Tao Zheng. 2023a. MixEdit: Revisiting data augmentation and beyond for grammatical error correction. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10161–10175, Singapore. Association for Computational Linguistics.

Jingheng Ye, Yinghui Li, Shirong Ma, Rui Xie, Wei Wu, and Hai-Tao Zheng. 2022. Focus is what you need for chinese grammatical error correction. *arXiv preprint arXiv:2210.12692*.

Jingheng Ye, Yinghui Li, and Haitao Zheng. 2023b. System report for CCL23-eval task 7: THU KELab (sz) - exploring data augmentation and denoising for

Chinese grammatical error correction. In *Proceedings of the 22nd Chinese National Conference on Computational Linguistics (Volume 3: Evaluations)*, pages 262–270, Harbin, China. Chinese Information Processing Society of China.

Jingheng Ye, Yinghui Li, Qingyu Zhou, Yangning Li, Shirong Ma, Hai-Tao Zheng, and Ying Shen. 2023c. CLEME: Debiasing multi-reference evaluation for grammatical error correction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6174–6189, Singapore. Association for Computational Linguistics.

Jingheng Ye, Shang Qin, Yinghui Li, Xuxin Cheng, Libo Qin, Hai-Tao Zheng, Ying Shen, Peng Xing, Zishan Xu, Guo Cheng, et al. 2025a. Excgec: A benchmark for edit-wise explainable chinese grammatical error correction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25678–25686.

Jingheng Ye, Shang Qin, Yinghui Li, Hai-Tao Zheng, Shen Wang, and Qingsong Wen. 2025b. Corrections meet explanations: A unified framework for explainable grammatical error correction. *arXiv preprint arXiv:2502.15261*.

Jingheng Ye, Shen Wang, Deqing Zou, Yibo Yan, Kun Wang, Hai-Tao Zheng, Zenglin Xu, Irwin King, Philip S Yu, and Qingsong Wen. 2025c. Position: Llms can be good tutors in foreign language education. *arXiv preprint arXiv:2502.05467*.

Jingheng Ye, Zishan Xu, Yinghui Li, Linlin Song, Qingyu Zhou, Hai-Tao Zheng, Ying Shen, Wenhao Jiang, Hong-Gee Kim, Ruitong Liu, Xin Su, and Zifei Shan. 2025d. CLEME2.0: Towards interpretable evaluation by disentangling edits for grammatical error correction. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 204–222, Vienna, Austria. Association for Computational Linguistics.

Zihao Yi, Jiarui Ouyang, Yuwen Liu, Tianhao Liao, Zhe Xu, and Ying Shen. 2024. A survey on recent advances in llm-based multi-turn dialogue systems. *arXiv preprint arXiv:2402.18013*.

Miao Yu, Junyuan Mao, Guibin Zhang, Jingheng Ye, Junfeng Fang, Aoxiao Zhong, Yang Liu, Yuxuan Liang, Kun Wang, and Qingsong Wen. 2024a. Mind scramble: Unveiling large language model psychology via typoglycemia. *arXiv preprint arXiv:2410.01677*.

Tianyu Yu, Chengyue Jiang, Chao Lou, Shen Huang, Xiaobin Wang, Wei Liu, Jiong Cai, Yangning Li, Yinghui Li, Kewei Tu, et al. 2024b. Seqgpt: An out-of-the-box large language model for open domain sequence understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19458–19467.

Hamed Zamani, Susan Dumais, Nick Craswell, Paul Bennett, and Gord Lueck. 2020. Generating clarifying questions for information retrieval. In *Proceedings of the web conference 2020*, pages 418–428.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Tong Zhang, Peixin Qin, Yang Deng, Chen Huang, Wenqiang Lei, Junhong Liu, Dingnan Jin, Hongru Liang, and Tat-Seng Chua. 2024. Clamber: A benchmark of identifying and clarifying ambiguous information needs in large language models. *arXiv preprint arXiv:2405.12063*.

Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W Bruce Croft. 2018. Towards conversational search and recommendation: System ask, user respond. In *Proceedings of the 27th acm international conference on information and knowledge management*, pages 177–186.

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

Deqing Zou, Jingheng Ye, Yulu Liu, Yu Wu, Zishan Xu, Yinghui Li, Hai-Tao Zheng, Bingxu An, Zhao Wei, and Yong Xu. 2025. Revisiting classification taxonomy for grammatical errors. *arXiv preprint arXiv:2502.11890*.

A Experimental Details

A.1 NER Pre-processing

To accomplish the retrieval process consistently for different product categories, we initially applied named entity recognition (NER) (Yu et al., 2024b; Li et al., 2024a; Tan et al., 2023) to the documents obtained from AliMe KG. We correct possible grammatical errors (Ye et al., 2023c,a; Huang et al., 2023a; Ye et al., 2022, 2023b, 2025d) in the original data before NER preprocessing. For this paper, we chose to use an e-commerce-specific NER model (Wang et al., 2021) trained with a Co-operative Learning objective. This coaching strategy encourages two input views to generate similar contextual representations or output label distributions. The NER processing transforms all data from various categories into structured documents with a consistent named entity schema consisting of 54 hierarchical entity labels. We then compress this label space into a new one composed of only 10 entity labels, which allows the agent to retrieve documents more efficiently and accurately from the SQL database. The details of the product item description are provided in Table 5. We also showcase several processed product items in Table 6.

A.2 Details about Retrievers

BM25. As the representative of sparse retrievers, BM25 (Robertson et al., 2009) consistently exhibits impressive performance across diverse information retrieval benchmarks (Long et al., 2022).

GTE. The General Text Embedding (GTE) model is a general-purpose text embedding model trained with the multi-stage contrastive learning objective (Li et al., 2022, 2023b). It has shown exceptional results in the Massive Text Embedding Benchmark (Muennighoff et al., 2023).

CoROM. The CoROM model (Long et al., 2022) uses BERT-base (Devlin et al., 2019) as its backbone. It is a dual Encoder-based embedding model (Karpukhin et al., 2020) trained on annotated query-passage e-commerce datasets.

Reranker. We also integrate the lightweight bge-reranker-base model (Xiao et al., 2023)[2] to rerank the top-k documents retrieved.

[2]https://huggingface.co/BAAI/bge-reranker-base

Name	Description	Type
Category	Category of the item.	str
Brand	Brand of the item.	List[str]
Series	Series of the item.	List[str]
Target Customer	Target customers of the item.	List[str]
Applicable Scenario	Applicable scenarios of the item.	List[str]
Decorative Attribute	Decorative attributes of the item.	List[str]
Material	Material of the item.	List[str]
Style	Styles of the item.	List[str]
Specification	Available specification of the item.	List[str]
Color	Available colors of the item.	List[str]
Function	Function of the item.	List[str]

Table 5: Product item description.

Fusion. Furthermore, we also seek to investigate whether the fusion of multiple heterogeneous retrievers can boost performance on our benchmark. Consequently, we test hybrid fusion retrievers that utilize the reciprocal rerank fusion algorithm (Cormack et al., 2009) without requiring any additional models or significant computation.

A.3 Product Aspects of Clarification Questions

We observe that most clarification questions focus on one specific product aspect indicated in Table 5, such as category, brand, and series. Therefore, we analyze the proportion of product facets to which clarification questions are related. The results, illustrated in Figure 5, show that all three LLMs tend to generate clarification questions with even distributions. To detail, three aspects, namely applicable scenarios, style, and function, respectively, account for more than 10% across LLMs. On the other hand, LLMs are less likely to clarify series and specifications, which may be too specialized for users. This reveals that LLMs naturally have a preference for asking specific types of clarification questions in this task since we do not incorporate any explicit constraints on clarification question generation.

B Prompts and Examples

B.1 Prompts for ProductAgent

We report all the prompts used in our proposed ProductAgent, including Text2SQL in Table 7, Query Generation in Table 8, and Clarification Question Generation in Table 9.

B.2 Prompt for User Simulator

We list the prompt for user simulation in Table 10.

Category: T-shirt **Title**: 2023 Domestic Purchase Authentic Counter SEIFINI Summer Style 3E4200571M Original **Brand**: ["SEIFINI"], **Series**: [], **Target Customer**: ["Female", "Youth"], **Applicable Scenario**: ["Spring"], **Decorative Attribute**: ["Counter", "Authentic"], **Material**: ["Other Materials"], **Style**: ["Summer Style"], **Specification**: ["M/160", "L/160" "XL/170", ...], **Color**: ["White"], **Function**: []
Category: Plastic Blocks **Title**: LEGO Easter Basket Rabbit Children's Building Educational Toy Festival Limited Edition New **Brand**: ["LEGO"], **Series**: [], **Target Customer**: ["Age group: <14 years", "Children"], **Applicable Scenario**: ["Easter", "Festival"], **Decorative Attribute**: ["Basket", "Rabbit", "Educational"], **Material**: [], **Style**: ["Building", "Plug-in"], **Specification**: ["40587"], **Color**: [], **Function**: []
Category: Sneakers **Title**: Japan Ulovazn Kids Shoes Children's Sneakers 2023 Spring New Simple Versatile Boys and Girls Little White Shoes **Brand**: ["Ulovazn"], **Series**: [], **Target Customer**: ["Male", "Kids", "Female", "Boys and Girls", "Children"], **Applicable Scenario**: ["Shopping Mall", "Spring", "Spring and Autumn"], **Decorative Attribute**: ["Sole"], **Material**: ["Rubber], **Style**: ["Sport", "Casual", "New", "Versatile", "Simple"], **Specification**: ["X23S2123", "Size 35", "Size36", ...], **Color**: ["White", "Beige", "Black"], **Function**: ["Deodorant", "Non-slip"]

Table 6: Example cases of processed product items.

B.3 Examples of Conversation

Here we provide a detailed conversation example regarding “Canvas shoes" in Table 11. We can notice that ProductAgent first generates some critical clarification questions, such as the color, applicable scenarios, and functions, which are helpful to precisely identify the basic user demands. Subsequently, the agent probes for additional specifics like style, decorative attributes, and material. On the other hand, the user simulator, which is driven by GPT-3.5, occasionally offers responses that go beyond the provided options.

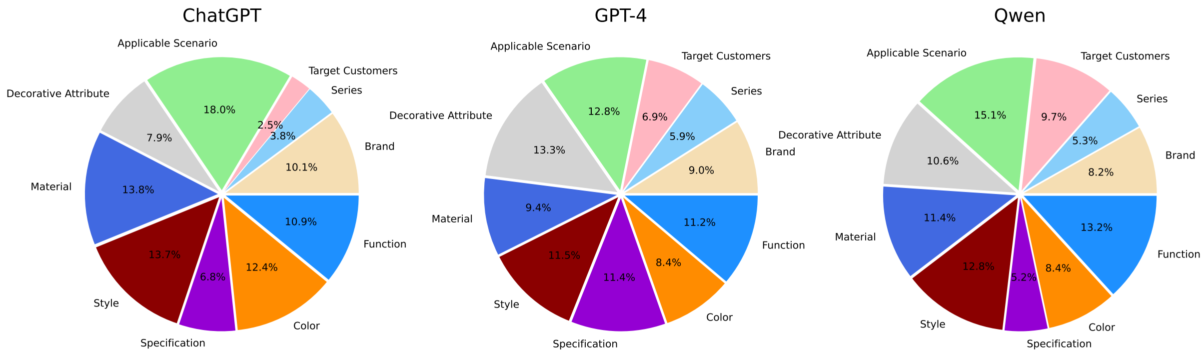

Figure 5: Proportion of product aspects of clarification questions generated by three LLMs.

Text2SQL Prompt Template

You are an SQL generation assistant. Given several constraints, you should generate a syntactically correct {dialect} SQL query statement to retrieve target records. To synthesize reasonable queries, you must follow the following rules:

1. Directly generate only SQL query statements, without outputting any explanation or inference information.
2. Directly use "*" to retrieve all columns.
3. Use the keyword "LIMIT" to limit the maximum number of retrieved records to {max_number}.
4. Carefully construct the where conditions for the query and use the keyword 'LIKE' as much as possible.

Given the following table structure description, only retrieval of that table is allowed:
Table schema: {schema}

Example input:
Product category: Casual pants
Question: What material is the main material for the casual pants you would like to purchase?
Answer: Polyester fiber

Example output:
SQL Query: SELECT * From item WHERE category='Casual pants' AND material LIKE '% polyester fiber%' LIMIT {max_number};

Input:
Product category: {category}
Question: {question_1}
Answer: {answer_1}
...

SQL Query:

Table 7: The one-shot prompt template of the Text2SQL tool in ProductAgent.

Query Generation Prompt Template
You are a query generation assistant. Given the user's purchasing demands, you need to generate a short natural language query statement (Query) to retrieve the target product. In order to generate a reasonable query, you must follow the following rules: 1. The generated query should be concise, composed of keywords, and separated by spaces. 2. The generated query should cover all of the user's purchasing requirements. 3. Do not output any explanations or inference information, and do not use unnecessary punctuation such as quotation marks. **Product category**: {category} **Question**: {question_1} **Answer**: {answer_1} ... **Query**:

Table 8: The zero-shot prompt template of the Query Generation tool in ProductAgent.

Clarification Question Generation Prompt Template
You are a product shopping assistant that can accurately identify user demands, and you are capable of generating three multiple-choice questions for demand clarification. To help you ask valuable questions, here is a summary of statistics about {category}. **Statistics**: {statistics} 1. The generated content must focus on the product category ({category}) and contribute to accurately identifying user demands. 2. It is prohibited to generate new questions that are duplicates of previous ones. 3. When constructing options, try to directly select them from the statistical data. You need to generate new multiple-choice questions for demand clarification based on the historical Q&A. Here is the historical Q&A: **Question**: {question_1} **Answer**: {answer_1} ... The generated result should strictly follow the JSON format, and no additional information should be included. **JSON format description**: JSON_description ```JSON

Table 9: The zero-shot prompt template of the Clarification Question Generation tool in ProductAgent. We require LLM to synthesize clarification questions in structured JSON form.

User Simulator Prompt Template
You are in a conversation with a shopping assistant, hoping that they can help you search for suitable products. To provide reasonable answers, you must follow these rules: 1. You must answer the question accurately based on the target product. 2. Try to use the options provided for your answer, if your answer is not covered by the options, simply answer "Other". 3. Directly output the answer, do not provide any explanation or reasoning information and avoid unnecessary punctuation like quotation marks. 4. Each question's answer should be on a separate line. Here is the target product information: {item} Here is the official question from the assistant: {questions} Answer the above question directly, do not repeat the question, **each answer should be on a separate line**.

Table 10: The zero-shot prompt template of the user simulator.

Conversation example by GPT-4 driven ProductAgent and the user simulator

A: Hi, I'm here to assist you with product sales. Which category of the product are you looking for?

U: Canvas shoes.

A: Great! I've found several options for you: <Item$_1$>, <Item$_2$>, $\cdots$. To help me recommend the best choices, could you answer a few questions?
Q1: When you need canvas shoes, which color do you mainly focus on?
C1: ["Light gray", "Sky blue", "Yellow", "Military Green", "Rose Red", "Other"]
Q2: Which applicable scenarios do you prefer when wearing canvas shoes?
C2: ["Travel", "Dance", "Fitness", "Construction Site", "Official Flagship", "Other"]
Q3: When choosing canvas shoes, what functions do you value?
C3: ["Anti-slip", "Increase by 10CM", "Waterproof bottom", "Anti puncture", "Antibacterial", "Other"]

U: **Answers for Q1**: "Military Green". # Answering by selecting.
Answers for Q2: "Travel". # Answering by selecting.
Answers for Q3: "Other". # Answering by selecting.

A: I have search several relevant items for you: <Item$_1$>, <Item$_2$>, $\cdots$. Also, you can answer these questions so that I could recommend more suitable items for you!
Q1: When choosing canvas shoes, which style do you prefer?
C1: ["Leisure", "Internal elevation", "Ties", "Flat Heel", "Low help", "Other"]
Q2: What upper decoration attributes do you usually pay attention to?
C2: ["Adhesive", "Breathable", "Inside", "Mesh surface", "Sole", "Other"]
Q3: When purchasing canvas shoes, which material do you prefer to make them from?
C3: ["Mesh", "Rubber", "Canvas", "Other"]

U: **Answers for Q1**: "Leisure". # Answering by selecting.
Answers for Q2: Coffee leopard print, leopard print, round head. # Answering by typing.
Answers for Q3: "Canvas". # Answering by selecting.

A: I have search several relevant items for you: <Item$_1$>, <Item$_2$>, $\cdots$. Also, you can answer these questions so that I could recommend more suitable items for you!
...

Ground truth product item

Category: Canvas shoes
Title: Feiyue Thick Sole Canvas Shoes Women 2023 Spring Leopard Print Bread Shoes Casual Versatile Mantou Shoes
Brand: ["Feiyue"], **Series**: ["Spring 2023], **Target Customer**: ["Female", "18-40 years old", "Youth"], **Applicable Scenario**: ["Spring"], **Decorative Attribute**: ["Thick-soled", "Cross straps", "Coffee leopard print", "Shallow mouth", "Leopard print", "Pattern", "Roundhead", "Khaki leopard print"], **Material**: ["Rubber", "Canvas"], **Style**: ["Lace-up", "Low-cut", "Leopard print", "Casual", "Bread", "Versatile", "Vulcanization process"], **Specification**: ["EUR34", "EUR35", "EUR36", "EUR37", "EUR38", ...], **Color**: ["Purple", "Lake blue and orange", "Army green and white", "Pink", "Mixed colors"], **Function**: []

Table 11: A conversation example between ProductAgent (A) and the user (U). We also list the ground truth product item that the user is searching for.