Building Multi-Turn RAG for Customer Support with LLM Labeling

Zhiyu Chen
zhiyuche@amazon.com
Amazon.com, Inc.
Seattle, WA, USA

Biancen Xie
biancen@amazon.com
Amazon.com, Inc.
Seattle, WA, USA

Sidarth Srinivasan
srinisid@amazon.com
Amazon.com, Inc.
Seattle, WA, USA

Qun Liu
qunliu@amazon.com
Amazon.com, Inc.
Seattle, WA, USA

Manikandarajan Ramanathan
mramnat@amazon.com
Amazon.com, Inc.
Seattle, WA, USA

Rajashekar Maragoud
maragoud@amazon.com
Amazon.com, Inc.
Seattle, WA, USA

Abstract

Customer service in e-commerce often relies on human agents to handle inquiries related to orders, returns, and product information. While this approach is effective, it can be expensive and difficult to scale during periods of high demand. Recent advances in intelligent chatbots, particularly those based on Retrieval Augmented Generation (RAG) models, have significantly improved customer service efficiency by combining large language models with external knowledge sources. In the context of e-commerce, these systems can access up-to-date information from order databases, product catalogs, and support documents to manage complex, multi-turn interactions with customers. However, developing a multi-turn RAG chatbot for real-world customer service introduces additional challenges such as adaptive retrieval and query reformulation across dialogue turns. These components typically require large volumes of annotated data, which are often unavailable. To address this limitation, we propose methods that leverage large language models to automatically generate labels from real customer-agent dialogues. Specifically, we introduce two LLM-assisted labeling strategies for adaptive retrieval: an intent-guided strategy and an explanation-based strategy. For query reformulation, we explore two approaches: natural language reformulation and keyword-based reformulation. Our experiments show that the explanation-based strategy achieves the best results for adaptive retrieval, while keyword-based reformulation improves the quality of retrieved documents. These findings provide practical insights for developing scalable and intelligent customer support solutions in the e-commerce industry.

CCS Concepts

• **Information systems** → **Query reformulation**; **Query intent**.

Keywords

retrieval-augmented generation, query reformulation

ACM Reference Format:
Zhiyu Chen, Biancen Xie, Sidarth Srinivasan, Qun Liu, Manikandarajan Ramanathan, and Rajashekar Maragoud. 2025. Building Multi-Turn RAG for Customer Support with LLM Labeling. In *Proceedings of LLM4ECommerce Workshop at the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (LLM4ECommerce Workshop at KDD '25).* ACM, New York, NY, USA, 11 pages.

1 Introduction

Traditional customer service operations in e-commerce often rely on human agents to handle customer inquiries, leading to high operational costs and slower response times. In recent years, intelligent customer service chatbots [3, 6, 6, 15] have reshaped customer support by automating responses and improving efficiency. Among these advancements, Retrieval-Augmented Generation (RAG) [17] has emerged as a powerful technique for enhancing question-answering (QA) ability of customer service chatbots. By integrating large language models (LLMs) with external knowledge retrieval, RAG-based chatbots generate more accurate and contextually relevant responses [7, 9].

Compared to single-turn RAG-based QA systems, building multi-turn RAG-based chatbots [8, 16] for real-world customer service requires two additional components. The **adaptive retrieval** component determines when retrieval is necessary, reducing both latency and context length by fetching documents only when needed. The **query reformulation** component processes conversation history to generate precise queries for the retrieval module, ensuring contextually relevant responses.

Building adaptive retrieval and query reformulation components typically requires a substantial amount of annotated data. To overcome this challenge, we develop methods that leverage large language models to automatically generate labels for both components using customer-agent service dialogues collected in compliance with data handling policies. We demonstrate that models trained on real human-to-human conversations, combined with LLM-generated labels, can effectively support the development of a fully functional multi-turn RAG chatbot for customer service.

To generate labels for adaptive retrieval, we propose two labeling strategies using LLMs. The first, an intent-guided labeling strategy, leverages pre-defined intents to direct the labeling process. The second, an explanation-based strategy, directly prompts the LLM to label whether retrieval is needed and generate reasonable explanations for the decision. We also propose two strategies for natural language query reformulation and keyword formulation. The former rewrites the customer utterance into a self-contained, decontextualized, and well-structured question, while the latter generates a keyword-based query. Through experiments, we find that the explanation-guided strategy generates the highest quality labels for adaptive retrieval. Additionally, we observe that the keyword-based strategy retrieves higher quality documents.

We summarize the contributions of our paper as follows:

- We propose two LLM-based labeling strategies for adaptive retrieval and two for query reformulation to support the development of multi-turn RAG systems.
- Our experiments demonstrate that the explanation-guided labeling strategy is the most effective for generating adaptive retrieval labels.
- We also show that the keyword-based reformulation labeling strategy is more effective for training query reformulations that retrieve higher-quality documents, even though the reformulations may not be as fluent as natural language query reformulations.
- Our labeling strategies and experimental results provide valuable insights and guidance for practitioners in the industry working to build multi-turn RAG systems.

2 Related Work

2.1 Retrieval-Augmented Generation (RAG)

Integrated with retrieval mechanisms [9], RAG improves factual grounding and reduce hallucination in question answering systems [17]. A key challenge for RAG in real-time applications is balancing retrieval quality with computational efficiency. Studies such as Mallen et al. [12] revealed that indiscriminate retrieval increases latency and can degrade performance when irrelevant passages overwhelm the generator. To address this, recent work focuses on adaptive retrieval strategies. For example, Yao et al. [21] proposed confidence-based retrieval, where the LM triggers retrieval only when its internal uncertainty exceeds a threshold. In conversational settings, Roy et al. [16] designed self-multi-RAG, where an LLM determines when retrieval is needed given the dialogue context, then rewrites the conversation into a query if needed and filters the retrieved passages before answering. Su et al. [18] proposes Dynamic RAG named DRAGIN, which actively decides when to trigger retrieval and what to retrieve during generation. Unlike static one-shot retrieval, DRAGIN monitors the LLM's internal information needs across the generation process to decide the optimal moment to retrieve and to craft an appropriate query.

2.2 Query Reformulation

Query reformulation plays a crucial role in Conversational Question Answering (CQA) and Conversational Search (CS) by refining user queries to improve retrieval effectiveness and response relevance. In CQA and CS, users often ask follow-up questions that omit context from previous turns, necessitating reformulation into explicit, self-contained queries. Traditional methods rely on rule-based heuristics or query expansion [2], while neural approaches [1, 13, 14, 19] use sequence-to-sequence models to incorporate contextual information. Reinforcement learning-based methods [4, 5, 20] further optimize reformulation by maximizing downstream QA performance.

RAG-based frameworks have recently emerged as powerful QA solutions. Since RAG pipelines rely on retrieved documents to generate responses, refining input queries is crucial for retrieving high-quality evidence [10, 11]. While existing research focuses primarily on single-turn RAG, multi-turn RAG remains underexplored due to the lack of benchmarks. Scaling multi-turn RAG for industry applications presents additional challenges, including adaptive retrieval prediction and conversational query reformulation, as discussed in §3. Instead of proposing new models, this paper introduces a method for leveraging existing dialogues between human agents and customers to generate labels for adaptive retrieval and query reformulation.

3 Preliminary

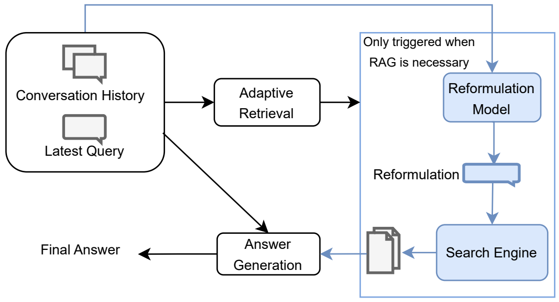

Figure 1: An overview of the adaptive retrieval and reformulation component integrated into RAG for multi-turn conversations.

In this section, we describe how adaptive retrieval and query reformulation are integrated into a RAG system to support multi-turn conversations. The overall framework is illustrated in Figure 1.

First, given the conversational history C and a customer's current utterance q as input, the adaptive retrieval model $\mathcal{M}_s$ first decides on whether to initiate an information retrieval process:

$$p_s = \mathcal{M}_s(C, q) \tag{1}$$

where $p_s = 1$ indicates retrieval is necessary otherwise $p_s = 0$. Here the input can be the concatenation of C and q.

If retrieval is not necessary, then the answer generation model directly generates the answer given C and q. If a retrieval is needed (the blue workflows in Figure 1), the reformulation model $\mathcal{M}_r$ will then rewrite the query:

$$q' = \mathcal{M}_r(C, q) \tag{2}$$

The reformulated query q' improves retrieval by resolving conversational dependencies and clarifying ambiguity with added context.

Considering the latency requirement, we use a foundation model such as Claude 3.5 Sonnet[1] only for the answer generation module in production, while keeping the other components as smaller models to minimize overall latency. Training $\mathcal{M}_s$ and $\mathcal{M}_r$ requires a large amount of data, which can be labor-intensive. However, as an e-commerce company, we have access to millions of real customer service transcripts containing dialogues between human agents and customers. In this work, we focus on a single key challenge: how to effectively leverage this valuable data, with the help of large language models, t, to train critical components (i.e., $\mathcal{M}_s$ and $\mathcal{M}_r$) for a multi-turn RAG chatbot capable of answering customer questions at scale.

[1] https://www.anthropic.com/news/claude-3-5-sonnet

4 Method

In real conversations, either the agent or the customer may consecutively input several utterances. However, the dialogue between a customer and a RAG-based chatbot is typically conducted in an alternating manner. To construct data that aligns with the chatbot format, we merge consecutive utterances from the same role into a single unit, resulting in a dialogue $d = [q_1, a_1, ..., q_n, a_n]$ where q_i represents the user's query and a_i represents the agent's response. For a query q_i, we define its context or conversation history as $C_i = [q_1, a_1, ..., q_{i-1}, a_{i-1}]$. In the following, we propose different label generation strategies for adaptive retrieval and query reformulation using LLMs with $d \in D$.

4.1 Labeling Strategy for Adaptive Retrieval

We consider two prompt strategies for adaptive retrieval labeling.

Intent-guided Labeling We instruct the LLM to classify the customer's intent before determining whether retrieval is necessary. The intent labeling is not directly used for model training. We defined 12 intents for customer queries which can be found in the prompt in Table 6. The underlying assumption is that the need for retrieval is strongly dependent on the query intent, and we hypothesize that prompting the LLM to reason through the query intent will enable it to more accurately assess whether retrieval is required.

Explanation-guided Labeling Instead of using pre-defined intents to guide the LLM annotation, we prompt the LLM to freely explain why it makes the decision that a retrieval is needed or not, and then generate the final label of p_s.

For a dialogue $d \in D$ with n customer utterances, rather than asking the LLM to annotate each query individually, we provide the entire dialogue to the LLM and ask it to output labels for all customer utterances. Interestingly, we find that this approach leads to a higher annotation accuracy compared to turn-level annotation, where each request includes only the context and the query rather than the full dialogue. We compare different annotation strategies in Section 6.1. The prompt templates for the two proposed strategies are shown in Table 6 and Table 7, respectively.

4.2 Labeling Strategy for Query Reformulation

To generate reformulation labels, we prompt an LLM with context and a customer query. We explore two annotation strategies.

NLQ Reformulation The first approach focuses on Natural Language Query (NLQ) reformulation, where the LLM generates a fully unambiguous query in natural language. This involves resolving any ambiguities in the original utterance, ensuring grammatical correctness, and addressing challenges such as co-reference resolution and omissions. During reformulation, we also prompt the LLM to explain the actions or edits it performed to generate the revised query.

Keywords Reformulation The second approach focuses on generating relevant keywords given the conversational context and user query. Compared to NLQ reformulation, keywords reformulation is not required to produce a fully structured natural language question. Instead, it extracts and prioritizes key terms that capture the essential intent of the user's query. We also prompt the LLM to explain the importance of the generated keywords for downstream retrieval.

While NLQ reformulations are commonly used in CQA [1, 5, 19], we find that they are not always necessary. Instead, keyword reformulation is sufficient to retrieve high-quality documents within the RAG framework. We present our findings in Section 6.3. The prompt templates for the two proposed strategies are shown in Table 8 and Table 9, respectively.

5 Experimental Setup

5.1 Datasets

We collected 10,000 real dialogues between agents and customers from the customer service of a well-known e-commerce website.

We use Claude-Sonnet-3 to generate labels for the training data using the methods proposed in Section 4. To protect customer privacy, all transcripts are scrubbed to exclude any personally identifiable information. Additionally, we mask the transcripts to remove any details pertaining to the company. We split the dataset into 80% for training and the remaining 20% for testing.

Table 1: Accuracy of LLM's annotation for adaptive retrieval.

Input Type	Labeling Strategy	Accuracy
full dialogue	Intent-guided	89%
full dialogue	Explanation-guided	92%
context + query	Intent-guided	85%
context + query	Explanation-guided	87%

Table 2: Evaluation result for adaptive retrieval

Input Type	Labeling Strategy	Accuracy	AUC	Ground Truth
context + query	Intent-guided	83.27%	92.49%	LLM
	Explanation-guided	86.49%	93.76%	LLM
	Intent-guided	74.27%	86.16%	Human
	Explanation-guided	83.39%	92.00%	Human
query	Intent-guided	82.54%	89.37%	LLM
	Explanation-guided	85.37%	92.41%	LLM
	Intent-guided	83.06%	91.93%	Human
	Explanation-guided	86.00%	92.82%	Human

5.2 Implementation Details

For adaptive retrieval, we consider two input types: one using only the user query and the other incorporating both the context and query as input. This help us understand whether incorporating conversation history improves the ability to identify when retrieving relevant information is beneficial. We use RoBERTa-base and optimize it with cross-entropy.

For query reformulation, we experiment with BART-base and FLAN-T5-base, fine-tuning the models to generate reformulations using LLM annotations as ground truth.

5.3 Evaluation Strategies

Adaptive Retrieval To evaluate the accuracy of the LLM in labeling for adaptive retrieval, we first assess different annotation strategies by manually examining 500 sampled test set. For adaptive retrieval models trained on labels from various annotation strategies, we report accuracy and AUC against LLM-generated labels, measuring how well the model distills knowledge from the labeled data. To further validate performance, we also evaluate the model on human-annotated data.

Reformulation To evaluate the performance of the query reformulation model, we assess its accuracy using automatic metrics, including BLEU and ROUGE, by comparing its reformulated queries against those generated by the LLM.

Compared to keyword reformulations, NLQ reformulations have stricter constraints, as they must be well-formed questions that resolve ellipses and co-references. A human evaluation presented in Appendix A shows that our method produces reformulations comparable to those generated by LLMs.

6 Results

6.1 Evaluation on Adaptive Retrieval

First, we evaluate the accuracy of different prompting strategies for generating adaptive retrieval labels, with results presented in Table 1. As shown, for the same input type, the explanation-guided method outperforms the intent-guided method, as some intents encompass both retrieval-needed and retrieval-not-needed cases. Additionally, dialogue-level annotation surpasses query-level annotation. We hypothesize that this improvement occurs because the LLM can better comprehend the queries when provided with the complete context of the dialogue, including future utterances.

To evaluate the effectiveness of the adaptive retrieval model, we train the model using labels generated with different strategies and input types, then test it on both LLM-generated and human-annotated labels. Based on the results in Table 2, we find that the accuracy gap between the synthetic dataset (generated by LLM) and the human-labeled dataset ranges from 1% to 6%, with higher accuracy on the synthetic dataset, likely due to the models being trained on synthetic data. On the human annotated test set, the explanation-guided labeling strategy outperforms intent-guided labeling, as it avoids predefined intents and instead allows the LLM to freely explain its predictions. Additionally, incorporating conversational history does not improve performance and, in some cases, even degrades it. Finally, models trained on LLM-generated data perform well on both synthetic and human-labeled datasets, with the best accuracy and AUC on human-labeled data achieved by the model trained using the explanation-guided labeling strategy without conversational history. This suggests that conversational context is not essential for adaptive retrieval within the scope of our datasets.

6.2 Evaluation on Query Reformulation

Automatic Evaluation Table 4 presents the automatic evaluation metrics. Both models perform similarly overall, but NLQ Reformulation yields higher BLEU and lower ROUGE-1 scores than Keyword Reformulation. This suggests NLQ Reformulations are more fluent and semantically coherent, with paraphrasing that reduces exact word matches. In contrast, Keyword Reformulation favors lexical overlap, retaining key terms at the expense of fluency and variation.

Table 3: The relevance scores for the Top-1 and Top-5 retrieval results using different reformulation strategies.

Query Type	Model	Top-1	Top-5
Query	-	0.76	1.07
	-	0.94	1.28
NLQ Reformulation	BART-base	1.26	1.43
	FLAN-T5-base	1.19	1.39
	Claude	1.31	1.5
Keywords Reformulation	BART-base	1.3	1.5
	FLAN-T5-base	**1.36**	1.51
	Claude	1.34	**1.59**

Table 4: Automatic evaluation result for query reformulation.

Reformulation Type	Model	BLEU	ROUGE-1	ROUGE-L
NLP Reformulation	BART-base	0.312	0.547	0.517
	FLAN-T5-base	0.305	0.550	0.520
Keywords Reformulation	BART-base	0.209	0.590	0.518
	FLAN-T5-base	0.204	0.592	0.520

6.3 Evaluation on Retrieval Performance

To evaluate the impact of reformulation on retrieval performance, we use various input types in our internal search engine to retrieve documents and employ LLM-based evaluation to measure the relevance of the top-k retrieved document to the context given a query. The reformulation model used in this evaluation is based on BART-base. LLM-based evaluation details are described in Table 10 in Appendix.

Table 3 presents the retrieval performance (Top-1 and Top-5) across different query types and models. While Claude was used to generate the reformulation ground-truth, our fine-tuned models based on those labels achieve competitive retrieval results. In fact, FLAN-T5-base outperforms Claude in Top-1 retrieval, demonstrating that our fine-tuned models can generate reformulations that yield better retrieval results in certain cases. Keywords reformulation consistently shows higher retrieval performance compared to NLQ reformulation across different models. This suggests that, even though NLQ reformulation is commonly used for conversational question answering, keywords reformulation is sufficient and can lead to better retrieval outcomes in a multiturn RAG setting.

7 Conclusion

We addressed the challenges of building a multi-turn RAG-based customer service chatbot by focusing on adaptive retrieval and query reformulation. To mitigate limited annotated data, we proposed automatic labeling methods using real customer-agent dialogues. Explanation-guided labeling outperformed intent-guided labeling for adaptive retrieval, achieving 86% accuracy and 92.82% AUC. For query reformulation, NLQ reformulations were more fluent, while keyword-based reformulations improved retrieval, with FLAN-T5-base achieving a Top-1 relevance score of 1.36. Notably, conversational context was unnecessary for adaptive retrieval, highlighting the effectiveness of simpler, query-focused models. Our approach offers a practical path for developing RAG systems using existing dialogue data with minimal manual annotation.

References

[1] Raviteja Anantha, Svitlana Vakulenko, Zhucheng Tu, Shayne Longpre, Stephen Pulman, and Srinivas Chappidi. 2021. Open-Domain Question Answering Goes Conversational via Question Rewriting. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (Eds.). Association for Computational Linguistics, Online, 520–534. doi:10.18653/v1/2021.naacl-main.44

[2] Peter Anick. 2003. Using terminological feedback for web search refinement: a log-based study. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. 88–95.

[3] J Benita, Kosireddy Vivek Charan Tej, E Vinay Kumar, G Venkata Subbarao, and CH Venkatesh. 2024. Implementation of Retrieval-Augmented Generation (RAG) in Chatbot Systems for Enhanced Real-Time Customer Support in E-Commerce. In *2024 3rd International Conference on Automation, Computing and Renewable Systems (ICACRS)*. IEEE, 1381–1388.

[4] Christian Buck, Jannis Bulian, Massimiliano Ciaramita, Wojciech Gajewski, Andrea Gesmundo, Neil Houlsby, and Wei Wang. 2018. Ask the Right Questions: Active Question Reformulation with Reinforcement Learning. In *International Conference on Learning Representations*. https://openreview.net/forum?id=S1CChZ-CZ

[5] Zhiyu Chen, Jie Zhao, Anjie Fang, Besnik Fetahu, Oleg Rokhlenko, and Shervin Malmasi. 2022. Reinforced Question Rewriting for Conversational Question Answering. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*, Yunyao Li and Angeliki Lazaridou (Eds.). Association for Computational Linguistics, Abu Dhabi, UAE, 357–370. doi:10.18653/v1/2022.emnlp-industry.36

[6] Lei Cui, Shaohan Huang, Furu Wei, Chuanqi Tan, Chaoqun Duan, and Ming Zhou. 2017. Superagent: A customer service chatbot for e-commerce websites. In *Proceedings of ACL 2017, system demonstrations*. 97–102.

[7] N. Ding, X. Xie, and Y. Feng. 2022. A Survey on Retrieval-Augmented Generation. *arXiv preprint arXiv:2202.01110* (2022).

[8] Yannis Katsis, Sara Rosenthal, Kshitij Fadnis, Chulaka Gunasekara, Young-Suk Lee, Lucian Popa, Vraj Shah, Huaiyu Zhu, Danish Contractor, and Marina Danilevsky. 2025. MTRAG: A Multi-Turn Conversational Benchmark for Evaluating Retrieval-Augmented Generation Systems. arXiv:2501.03468 [cs.CL] https://arxiv.org/abs/2501.03468

[9] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, ..., and S. Riedel. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems*.

[10] Zhicong Li, Jiahao Wang, Hangyu Mao, ZhiShu Jiang, Zhongxia Chen, Du Jiazhen, Fuzheng Zhang, Di ZHANG, and Yong Liu. [n. d.]. DMQR-RAG: Diverse Multi-Query Rewriting in Retrieval-Augmented Generation. ([n. d.]).

[11] Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query Rewriting for Retrieval-Augmented Large Language Models. *arXiv preprint arXiv:2305.14283* (2023).

[12] Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When Not to Trust Language Models: Investigating Effectiveness of Parametric and Non-Parametric Memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 9802–9822. doi:10.18653/v1/2023.acl-long.546

[13] Fengran Mo, Abbas Ghaddar, Kelong Mao, Mehdi Rezagholizadeh, Boxing Chen, Qun Liu, and Jian-Yun Nie. 2024. CHIQ: Contextual History Enhancement for Improving Query Rewriting in Conversational Search. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, Miami, Florida, USA, 2253–2268. doi:10.18653/v1/2024.emnlp-main.135

[14] Fengran Mo, Kelong Mao, Yutao Zhu, Yihong Wu, Kaiyu Huang, and Jian-Yun Nie. 2023. ConvGQR: Generative Query Reformulation for Conversational Search. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 4998–5012. doi:10.18653/v1/2023.acl-long.274

[15] Haode Qi, Lin Pan, Atin Sood, Abhishek Shah, Ladislav Kunc, Mo Yu, and Saloni Potdar. 2021. Benchmarking Commercial Intent Detection Services with Practice-Driven Evaluations. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers*, Young-bum Kim, Yunyao Li, and Owen Rambow (Eds.). Association for Computational Linguistics, Online, 304–310. doi:10.18653/v1/2021.naacl-industry.38

[16] Nirmal Roy, Leonardo F. R. Ribeiro, Rexhina Blloshmi, and Kevin Small. 2024. Learning When to Retrieve, What to Rewrite, and How to Respond in Conversational QA. arXiv:2409.15515 [cs.CL] https://arxiv.org/abs/2409.15515

[17] John Smith, Emily Johnson, and Michael Lee. 2023. Retrieval-Augmented Generation with Knowledge Graphs for Customer Service Question Answering. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online and In-Person, 1235–1245.

[18] Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. 2024. DRAGIN: Dynamic Retrieval Augmented Generation based on the Real-time Information Needs of Large Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Bangkok, Thailand, 12991–13013. doi:10.18653/v1/2024.acl-long.702

[19] Svitlana Vakulenko, Shayne Longpre, Zhucheng Tu, and Raviteja Anantha. 2021. Question rewriting for conversational question answering. In *Proceedings of the 14th ACM international conference on web search and data mining*. 355–363.

[20] Zeqiu Wu, Yi Luan, Hannah Rashkin, David Reitter, Hannaneh Hajishirzi, Mari Ostendorf, and Gaurav Singh Tomar. 2022. CONQRR: Conversational Query Rewriting for Retrieval with Reinforcement Learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (Eds.). Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 10000–10014. doi:10.18653/v1/2022.emnlp-main.679

[21] Zijun Yao, Weijian Qi, Liangming Pan, Shulin Cao, Linmei Hu, Weichuan Liu, Lei Hou, and Juanzi Li. 2024. Seakr: Self-aware knowledge retrieval for adaptive retrieval augmented generation. *arXiv preprint arXiv:2406.19215* (2024).

Appendix

A Human Evaluation on NLQ Reformulation

Compared to keyword reformulations, NLQ reformulations have stricter constraints, as they must be well-formed questions that resolve ellipses and co-references. To assess the quality of NLQ reformulations generated by the foundation LLMs or models trained on LLM-labeled data, we conduct a human evaluation on 500 sampled test set. For this evaluation, we propose and assess three key metrics:

- **Grammatical Correctness**: Evaluates whether the reformulation is grammatically correct.
- **Context Carryover Completeness**: Assesses whether the reformulation is self-contained and understandable without requiring reference to the dialogue history.
- **Context Carryover Accuracy**: Determines whether omissions and co-references in the reformulation are resolved correctly.

Human Evaluation For NLQ reformulation, we further evaluate the reformulation quality based on 500 sampled testing set. The numbers are reported in Table 5. Overall, all models perform well across the three evaluation criteria. Claude achieves the highest scores in all metrics, demonstrating its strong capability in NLQ reformulation under our proposed labeling strategy. FLAN-T5-base outperforms BART-base in grammatical correctness (GC) compared to BART-base, indicating its strength in producing well-formed outputs. However, BART-base shows slightly better performance in context carryover accuracy (CCA) and context carryover completeness (CCC), suggesting that it more effectively preserves contextual information during reformulation.

Table 5: Human evaluation on query reformulation results.

Model	CCA	CCC	GC
BART-base	94%	94%	95%
FLAN-T5-base	91%	91%	97%
Claude	96%	96%	97%

B Prompts

System Prompt

Instruction

Your task is to annotate a conversation between a customer and an agent with pre-defined intents and whether retrieval is needed to answer the customer utterance. The transcript is made up of multiple turns and each turn has the format of “Turn_ID.Role: utterance”. For example, “24.customer: But I bought the whole album.” means in the 24th turn, the customer said “But I bought the whole album.” Please read the transcript carefully, as I will ask you to label the intent of each customer utterance and whether it requires an API call to a search engine to obtain more information from a knowledge base in order to answer the query.

For every customer utterance, select one from the following intents:
a: DESCRIBE AN ISSUE. the customer is describing the issue.
b: ASK AN ISSUE RELATED QUESTION. the customer asks an issue-related question.
c: ANSWER A QUESTION. the customer is answering a question asked by the agent.
d: CONFRIMATION. the customer is simply confirming or recognizing information without making a specific request or inquiry.
e: REPORT STATUS. customer is reporting his or her current status, The customer has received instructions and is providing an update on the current issue status based on those instructions.
f: REQUEST. customer is making a request such as refund.
g: GRATITUDE. the turn is spoken by the customer, and customer is showing gratitude to the agent or bot the customer is expressing appreciation or thanks
h: COMPLAIN. the customer is expressing frustration about a product, service, experience.
i: ASK AGENT STATUS. the customer is asking about the availability or status of a agent.
j: FAREWELL. a customer’s intention to end or conclude the conversation.
k: GREETINGS. the turn is spoken by the customer.
l: OTHERS.

After labeling each customer turn’s intent, you will also label whether that turn will trigger a RAG based on the following criteria:
Yes: the current turn is spoken by the customer and the current utterance mentioned about useful details about the issue and the knowledge bank should be updated by a retrieval action based on current utterance information and previous conversation history. For example, RAG should be triggered when the customer describes an issue or asks an issue-related question. Additionally, RAG may also be necessary when the customer provides more details in response to a question or reports their status. However, in cases where no valuable information is provided, RAG should not be triggered.
No: There is no need to update the knowledge bank since the utterance does not contain any issue-related information. Overall, for each turn, you will first give an intent label(a-l) and then give a RAG label You should note that the intent label is closely related to RAG, RAG probably needs to be triggered. Follow the example below to format your answers.

Format Examples

Transcript in the format of:
<Transcript> 1.agent: Hello, welcome back. How can I help with? 2.customer: my remote is not working. 3.agent:Thanks, I apologize for the inconvenience, can you try restarting your tv 4.customer: Sure.</Transcript>
Output in the format of:
<Labels for intent and RAG>: 2.a,Yes,4.d,No </Labels for intent and RAG>

User Prompt
<Transcript>: **{Dialogue}**
<Labels for intent and RAG>:

Table 6: Prompt for intent-guided adaptive retrieval labeling.

System Prompt

Instruction

Your task is to understand a conversation between a customer and an agent and predict, for each customer utterance, whether it should trigger an API call to a search engine to retrieve more information from a knowledge base to answer the customer query. You should also provide an explanation for your decision. I will provide you with a conversation transcript between a customer and an agent. The transcript is made up of multiple turns and each turn has the format of "Turn_ID.Role: utterance". For example, "24.customer: But I bought the whole album." means in the 24th turn, the customer said "But I bought the whole album."

1. Explain the reason why an API call to trigger retrieval is needed or not needed for the utterance.

- Specifically, a search engine API call is needed if:

The utterance is an description about the issue. The utterance provides context regarding the issue, such as device, or subscription information. The utterance provides answers to an issue-related question or clarification question asked by the agent. The utterance confirms details or status regarding the device, plan or issue. The utterance asks questions (usually start with "how")regarding how to resolve a problem or how to complete certain steps. The utterance provides the status of the issue after trying something. The utterance is a continuation of the previous turn where retrieval is needed or recommended. Remember, if the mentioned issue details or context is not new, you could also consider retrieval is needed.

- A search engine API call is not needed if the utterance does not contain any issue-related information. For example, the utterance is a greeting, a farewell, a complain regarding the issue. confirming the issue is resolved.

2. Select a label from [Yes, No] to indicate whether an API call to retrieval is required for the utterance.

Format Examples

Transcript in the format of

<Transcript> 1.agent: Hello, welcome back. How can I help with? 2.customer: my remote is not working. 3.agent: Thanks, I apologize for the inconvenience, can you try restarting your tv 4.customer: Sure.</Transcript>

Output in the format of:

<Turn>[2,4]</Turn>

<explain> [2.The customer is describing an issue and retrieval is needed 4.the customer is expressing acknowledgement and the retrieval is not needed] </explain>

<label>[2.Yes,4.No]</label>

User Prompt

<Transcript>: **{Dialogue}**

<Annotation>

Table 7: Prompt for explanation-guided adaptive retrieval labeling

System Prompt

Instruction

I will provide you with a user query along with a conversation history between a customer and an agent. The conversation history consists of multiple turns, each numbered sequentially (e.g., 1 for turn 1, 2 for turn 2). Each turn begins with an indication of the speaker (e.g., if the agent is speaking, the turn will start with "agent:").

Your task is to analyze the entire conversation history along with the user query and generate a well-structured, clear, and optimized query that facilitates retrieving the most relevant information for a QA system. The reformulated query should be concise, precise, and designed to yield high-quality results. You can edit the query and perform the following action when reformulating the query:

Recover: Add missing context from previous conversation turns, such as resolving co-references.

Correction: Fix any grammatical or structural errors in the query.

Simplification: Exclude irrelevant information.

Before generating the reformulation, you need to first:

1. understand the query and describe the intent of the query.
2. explain what actions should be performed (Recover, Correction, Simplification) and why.
3. generate the reformulation.

Format Examples

Input in the format of

<Conversation history> 1.agent: What do you need help with? 2.customer: My Music App 3.agent: Sure, how can I help you with that?</Conversation history>

<Query>It does not work on my speakers</Query>

Output in the format of:

<Intent>issue description</Intent>

<Actions> replace the pronoun "it" with "Music App" mentioned in turn 2.</Actions>

<Reformulation>Music App does not work on my speakers.</Reformulation>

####Final Instruction

Do not change the intent of the original query. Do not change the query type of the original query. If the original query is a statement rather than a question, the reformulation should also be a statement. Keep the same tone, and make sure the reformulation sounds like something spoken by the customer to an agent. Make sure the reformulation is self-contained and can be understood without the conversation history.

User Prompt

<Conversation history>: **{Context}**

<Query>: **{Customer Query}**

<Output>:

Table 8: Prompt for NLQ reformulation generation

System Prompt

Instruction

I will provide you a user query along with a conversation history between a customer and an agent. The conversation history is made up of multiple turns, each numbered at the start. Your task is to take into account the entire conversation history along with the query to generate a new keywords query that will help retrieve the most relevant information. The reformulated query should be focus on core key phrases mentioned of the current turn and previous turns. Before generating the reformulation, you need to first:

1. understand the conversation history and current query, and then describe the intent of the query
2. explain why those keywords are generated, which turn are they coming from
3. generate the new keywords query

Format Examples

Input in the format of

<Conversation history>1.agent: What do you need help with? 2.customer: My Music App 3.agent: Sure, how can I help you with that?</Conversation history>

<Query>It does not work on my speakers</Query>

Output in the format of:

<Intent>issue description</Intent>

<Reason>The customer is mentioning that the music App is not working well on the speaker. Therefore the key information in the issue is music App, not work and Speakers</Reason>

<Reformulation> music App not work on speakers</Reformulation>

Final Instruction

Remember,

- The reformulation should be keywords.
- Each keyword could either be directly extracted from the conversation history and the current query, or relevant new keywords that could provide complementary information.

User Prompt

<Conversation history>: **{Context}**

<Query>: **{Customer Query}**

<Output>:

Table 9: Prompt for keywords reformulation generation.

System Prompt
Instruction
Given a list of documents, assess their relevance to a customer query and previous chat history between an agent and a customer. Each document consists of two fields: title and content. Rate the relevance on a scale from 0 to 2 based on its connection to the dialogue:
0: irrelevant, the document is irrelevant to the issues, product or services discussed in the dialogue
1: somewhat relevant, the document is related to the issues, product or services discussed in the dialogue
2: Perfectly relevant, the document contains critical information to address the query with comprehensive information

Format Examples
Input in the format of
<Chat History>[Here is the chat history]</Chat History>:
<Query>[Here is the customer query]</Query>:
Document:
<title>[Title of the document]</title>
<content>[Content of the document]</content>
Output in the format of:
<Score> select from 0-2
<Explanation>Provide a brief explanation of why you give this rating. Point out any specific areas where the document succeeds or fails in addressing the query.

User Prompt
<Chat History>: **{Context}**</Chat History>
<Query>**{Query}**</Query>:
Document:
<title>**{title}**</title>
<content>**{content}**</content>
Output:

Table 10: Prompt for LLM-based relevance judgment of the top-1 search result. The prompt for the top-5 results follows a similar structure but includes five documents.