Inquiry Assistant Using LLM-Generated Knowledge Graphs

István Varga
Recruit Co., Ltd. Megagon Labs
Tokyo, Japan
istvan@megagon.ai

Yuta Yamashita
Recruit Co., Ltd. Megagon Labs
Tokyo, Japan
y.yamashita@megagon.ai

Figure 1: Overview of our knowledge graph based proposal (above) versus the traditional RAG-LLM framework (below).

Abstract

Businesses are increasingly overwhelmed by inquiries related to their services or products. Relying on human agents to handle inquiries via email results in higher costs and delayed responses, contributing to customer dissatisfaction. In response to these challenges, this pilot study leverages advancements in Large Language Models (LLMs) by proposing a fully automated method for generating a knowledge graph from unstructured data in help pages, which is then utilized to power a fully automated dialogue management system. By transitioning to a chat-based approach, our method aims to handle ambiguous, incomplete, or nonspecific inquiries more effectively and enhance customer satisfaction with tailored, natural responses. We also implement explicit safeguards to improve intent identification and prevent response hallucinations. We validate our proposal in the hotel industry, demonstrating that our knowledge graph based AI agent outperforms the baseline Retrieval-Augmented Generation (RAG) model in accuracy while facilitating more natural and coherent dialogues.

CCS Concepts

• **Information systems → Document representation**; **Users and interactive retrieval**; • **Computing methodologies → Natural language processing**.

Keywords

Automated Customer Support, Conversational Agent, Knowledge Graph Generation, Automated Dialogue Management

1 Introduction

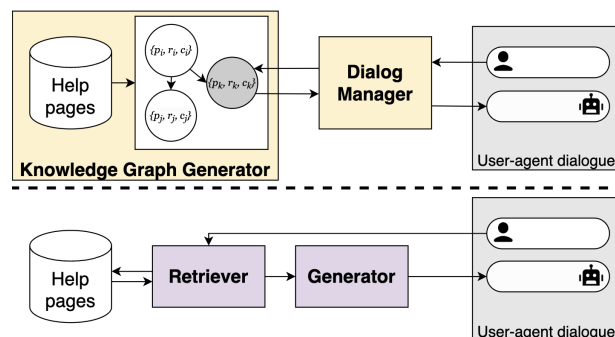Organizations are experiencing a growing volume of inquiries related to their services or products, making traditional manual email-based responses increasingly challenging. This leads to higher operational costs and delayed responses, ultimately contributing to customer dissatisfaction.

Recent advancements in large language models (LLMs) have significantly accelerated development in conversational AI [2, 3, 5, 9], particularly automated customer support [6, 10], making it possible to enhance user engagement at a reduced operational cost. Furthermore, advances in graph acquisition using LLMs [7, 8, 10–12] can further improve the expertise of such systems.

In our services, user inquiries often exhibit incompleteness or unspecificity. We anticipate that employing a dialogue-based interface will lead to more efficient resolutions. To illustrate this challenge, consider the following sample inquiries:

(1) *"I'm getting an error when trying to change my booking."*
(2) *"I can't find the confirmation code of my reservation."*

In Example 1, the lack of a detailed error description requires the agent to clarify the specific issue. In Example 2, the user does not specify the reservation method (e.g., online or by phone), which is essential for the agent to provide accurate support.

Our research has two fundamental priorities. First, we aim to facilitate natural and coherent dialogues that incorporate clarifying questions. This approach allows the AI agent to efficiently extract information and provide accurate, targeted responses that reflect the individuality of each inquiry, moving beyond generic automated templates. Second, we aim to maximize accuracy in user interactions through explicit safeguards that ensure a correct understanding of user intents and facilitate appropriate responses when sufficient information is available, thereby preventing the AI agent from generating incorrect responses when information is insufficient.

In this pilot study, we propose a fully automated *knowledge graph generator* that organizes information from existing help pages. Additionally, we introduce a *dialogue manager* that leverages the structure of this knowledge graph to identify user intent and conduct a dialogue that leads to the resolution of inquiries (Figure 1).

Retrieval-augmented generation (RAG) [4] with knowledge graphs has been employed in a customer service QA framework [12]. Our

Table 1: English adaptation of a help page with the topic "Reservation change" (original in Japanese).

Reservation change

If you wish to rebook, be aware that cancellation fees may apply. [...] For details, check directly with the accommodation.

Changing your reservation depends on the method used and the items you wish to modify. Select your reservation method:

1. Online reservation as logged in member:
   1.1. Change the accommodation: [resolution set #1]
   1.2. Change the plan: [resolution set #2]
   [...]
2. Online reservation as guest user:

Changes to the reservation are not possible. Cancel your current reservation and make a new one. [...] For information on how to cancel your reservation, refer to the following link: *Cancel Reservation.*

3. Reservation by phone:

Contact the call center. If the call center is closed and you need immediate assistance, contact the accommodation directly.

---

work differs in that, due to the complexities of real-life customer inquiries, we propose a dialogue-based approach. Additionally, existing researches have explored the use of conversational agents to enhance customer service in e-commerce, employing manually curated knowledge bases and supervised models [6]. Our proposal aims to minimize operational costs through full automation. Moreover, these previous approaches assume that user inquiries are comprehensive and do not require clarification from the agent. In contrast, our proposal emphasizes addressing incompleteness and unspecificity in user inquiries.

2 Our Approach

2.1 Knowledge Graph Generator

We use online help pages as our source to generate a knowledge graph. The complexity of these help pages can vary, ranging from short descriptions of very specific topics to detailed explanations of more generic subjects. Table 1 exhibits such a typical help page[1].

We make the following assumptions about the help pages:

(1) All help pages are centered around one specific core topic, any other potential topics are sub-topics of the core topic.
(2) The relationship between topics and the organizational structure of a help page can be leveraged to facilitate a natural and coherent dialogue.
(3) Each topic is accompanied by a resolution, often with conditions that specify when that resolution is relevant.
(4) All links between help pages connect relevant topics.

Accordingly, we define the following concepts that are central to our knowledge graph generator:

- **Problem topic**[2] (p): Topics in the help page that correspond to potential issues faced by users.
- **Resolution** (r): Instructions or steps provided in the help page on how to address the problem p.
- **Condition** (c): Contextual factors or specific situations in which a particular resolution r is applicable.

We further define relationships between *problems*:

- **Child**: An intra-article relationship where a generic problem points to its specific sub-problem.
- **Reference**: An inter-article relationship between two problems, indicated by a pre-existing URL in the help page[3].

We define our knowledge graph as a directed graph, where nodes are sets of $\{p, r, c\}$ attributes and edges connect nodes whose *problem* attributes are in either **child** or **reference** relationships[4].

Our proposed LLM-based knowledge graph generator utilizes a single prompt to perform the following actions step-by-step for each help page:

- **Step 1:** Identify the *root node* representing the core problem p as a noun phrase (e.g., *"reservation change"*), along with its condition c (e.g., *"reservation by phone"*) and resolution r as a snippet from the original help page.
- **Step 2:** Identify child nodes of the root node, i.e., nodes whose problem p attributes are sub-problems of the root node problem. Also identify their corresponding conditions c and resolutions r.
- **Step 3:** For each identified child node, iterate through Step 2 to identify further child nodes.

For the resulting knowledge graph, we generate the following attributes for each node using additional prompts:

- **Intents** (i): To explicitly facilitate mapping user utterances to nodes, the problem p and resolution r are paraphrased as *intents* from the user's perspective (e.g., $p =$*"reservation change"* $\rightarrow i = \{$*"How do I change my reservation?"*, *"I can't change my reservation"*, ...$\}$).
- **Actionable** (a): To prevent ineffective agent responses, we introduce a flag that indicates whether the resolution r is actionable. This flag is marked as *true* if the user can resolve the problem p based solely on the resolution r. It is marked as *false* if the resolution r is non-actionable, vague, or has **reference** edges to other nodes.

2.2 Dialogue Manager

Our dialogue manager relies entirely on the knowledge graph to determine the agent's action type and response content, without using the help pages. The dialogue manager consists of the following steps, each executed by independent prompts:

(1) **User intent identification:** Identify the user's intent from their last utterance by matching it against the knowledge graph's intents i node attributes[5].
(2) **Agent response candidate generation:** For each identified intent attribute from Step 1, generate *response candidates* using the corresponding nodes in the knowledge graph. The details of this process are described below.
(3) **Agent response selection:** Select the most appropriate *response candidate* from Step 2, based on relevance to the identified user intent and contextual continuity.

---

[1]https://help.jalan.net/jln/s/article/000004537 (Accessed on February 19, 2025).
[2]Hereafter, referred to as **problem** for the sake of brevity.

[3]This edge directs to the *root node* of the referenced help page.
[4]Note that the resulting knowledge graph generated from all help pages may not be a connected graph. We consider this outcome natural, as not all topics are interrelated.
[5]For the first user utterance, the entire knowledge graph is the target. For subsequent utterances, the target is limited to the connected graph containing the previous intent.

Agent response candidate generation is performed in the following steps, handled by a single prompt:

(1) **Action type selection:** Select the response type from[6]:
- **Resolution:** The target node's actionable a attribute is *true*, and if the condition c is not empty, it is fulfilled.
- **Clarification:** The target node either has an unfulfilled condition c, or it has multiple **child** nodes.
- **Escalation:** The target node's actionable a attribute is *false* and the node has no outgoing connections.

(2) **Response formulation:** Formulate the agent's response according to the *action type* determined in Step 1 and information from condition c and resolution r attributes. To prevent hallucination, the LLM is explicitly instructed to adhere to the core information of these attributes, while being allowed to adjust only the wording for naturalness.

3 Experiments and Results

For our offline pilot study, we evaluated the performance of the proposed method using a set of paraphrased inquiries derived from the inquiry logs of our hotel reservation service[7].

The topics of these inquiries encompass a wide range of issues, from general inquiries to requests for assistance across various subjects, including reservation changes or cancellations, payment methods, reward points, login errors, registration issues and so on.

For all our experiments we utilize GPT-4[8] [1].

3.1 Data Used in the Experiments

A total of 1,139 inquiries, representing one month's worth of data[9], have been manually labeled with a `gold response`, prior to this study. This `gold response` is a detailed response that contains all necessary information to resolve the inquiry in question[10].

Not all inquiries can be resolved with the information from help pages. Our analysis found that only about 44.95% of inquiries had clear resolutions from help pages. The rest often involve complex scenarios or require detailed booking information, which is typically accessible only through specific user or booking IDs. For these cases, only generic responses may be available, and we expected our models to accurately identify such inquiries. Our experiments focused separately on these two types: **resolvable inquiries**, which can be addressed using information from help pages, and **escalation-required inquiries**, which necessitate human intervention.

3.2 Target Models

In our experiments, we compared our proposed method (**Graph-Agent**) to a baseline method based on Retrieval-Augmented Generation (**RAG-Agent**). RAG-Agent did not utilize the knowledge graph, instead, at each dialogue turn it formulated its responses directly from the help pages in the following steps:

(1) **Retriever:** Identify potentially relevant help pages by matching their embeddings against the embedding of the core user intent, extracted from the user utterances. Empirically we set the cosine similarity threshold to 0.4; if no qualifying pages were found, we selected the top 3 similar ones.

(2) **Generator:** Based on the pages retained during the **retriever** step, perform **action type selection**, similarly to the corresponding step of Graph-Agent, followed by **response formulation** according to the selected action type.

3.3 User-Agent Dialogue Simulation

To evaluate the target models, we simulated user-agent dialogues with each of the agents. For the user's first utterance, we used our paraphrased inquiries. For subsequent user utterances, we employed a **user-LLM** model, with instructions to perform the following tasks:

(1) **Action type selection:** End the dialogue if the agent's response is considered satisfactory from the user's perspective, or continue the dialogue if there are still unaddressed or new issues.

(2) **Response formulation:** Based on the selected action type, formulate the appropriate user response.

3.4 Resolvable Inquiries Evaluation

For each dialogue generated from **resolvable inquiries**, we compared the `gold response` with the responses produced by each of the agents. We utilized the following three-way categorical metric:

- `correct`: All *core information* from the `gold response` is conveyed in the agent's response, thus ensuring that the resolution provided to the user's intent is correct.
- `partial`: Some *core information* from the `gold response` is conveyed in the agent's response, but critical information is missing. While the response remains useful, it may lead to a suboptimal resolution.
- `incorrect`: Information conveyed in the agent's response contradicts the *core information* in the `gold response`, or the agent incorrectly assesses that the response cannot be formulated from the help pages. The user's intent is either unresolved or the provided resolution is incorrect.

We utilized an **evaluator LLM** to assess each dialogue according to the above evaluation metric. To validate the evaluator LLM, we randomly selected 100 simulated dialogues for both Graph-Agent and RAG-Agent, which were then evaluated by three human annotators. We found that the general trends were similar, with Graph-Agent outperforming RAG-Agent[11] (Figure 2). Additionally, inter-annotator agreement[12] was moderate among human annotators (0.52 to 0.61) and between humans and the evaluator LLM (0.39 to 0.63). The largest disagreement stemmed from the evaluator LLM's difficulty in distinguishing between *core* and *complementary* information, leading to a higher `partial` ratio compared to

---

[6]Note that these conditions are not strict, they serve as guidance for the LLM, allowing it flexibility to determine the best course of action.

[7]We anonymized and paraphrased the original inquiries for privacy purposes.

[8]We use Azure OpenAI API's *2024-05-01-preview* model. For embedding representation with RAG-Agent, we use *text-embedding-3-large*.

[9]All inquiries from November 2023.

[10]Our current email-based inquiry response scheme leads to verbose replies, including both essential **core information** and additional **complementary information** that, while relevant, is not critical. In a dialogue-based scenario, prioritizing core information is more effective, as concise responses enhance user engagement and clarity.

[11]Majority is defined as when two out of three annotators assign the same category to a dialogue. Values for no majority are omitted from the table.

[12]We employed Cohen's Kappa with quadratic weights, as mismatches between `correct` and `partial` are less critical than those between `correct` - `incorrect` and `partial` - `incorrect`.
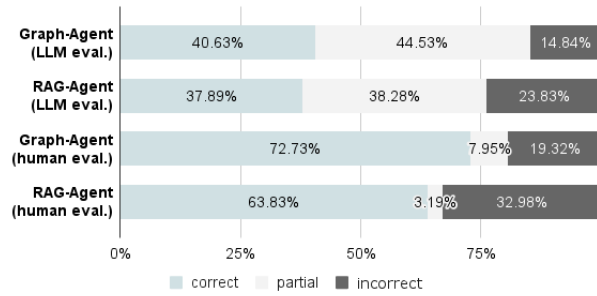
Figure 2: Model response accuracy for resolvable inquiries.

human evaluators. Both LLM and human evaluations indicate that Graph-Agent outperforms RAG-Agent in accuracy (Figure 2).

3.4.1 Dialogue Coherence. Graph-Agent demonstrated a higher tendency to ask clarifying questions, with 45.54% of its dialogues labeled as `correct` or `partial` including **clarification** actions, compared to 18.97% for RAG-Agent. This resulted in more dialogue turns (2.05 vs. 1.45). In contrast, RAG-Agent's responses, although often accurate, were frequently verbose, averaging 169 characters compared to Graph-Agent's 128, potentially obscuring resolution with excessive information.

3.4.2 Incorrect Responses. For resolvable inquiries expectation being a **resolution**, Graph-Agent produced fewer erroneous **escalation** actions compared to the RAG-Agent, with 12.11% of the total dialogues for the Graph-Agent versus 28.52% for the RAG-Agent, demonstrating the effectiveness of our **intents** attribute based intent identification. However, both models failed predominantly when user intents were unclear, causing intent recognition issues. Graph-Agent provided better control over responses, as resolutions and problems are attributes within the same knowledge graph node. In contrast, RAG-Agent frequently synthesized responses from multiple pages, leading to hallucinations characterized by contradictory or incorrect information. While Graph-Agent avoided hallucinations, it occasionally overlooked critical details, as help pages do not always position essential resolution information close to the corresponding problems, resulting in gaps in the knowledge graph.

3.5 Escalation-Required Inquiries Evaluation

We randomly selected 300 inquiries from and their corresponding dialogues from the **escalation-required** inquiry data. The expectation was that the models would assess them as not resolvable using the available information from the help pages.

The **escalation** ratio for both Graph-Agent and RAG-Agent was similar at 46% and 44%, respectively. We asked two human annotators to evaluate the remainder of the dialogues where the agents attempted to resolve the inquiry, using the following criteria:

- `useful`: The agent's response is valuable by providing a generic resolution or escalating the inquiry to relevant contacts, (e.g., hotel, service provider, or credit card company).
- `incorrect`: The agent's response is incorrect and does not lead to a resolution.
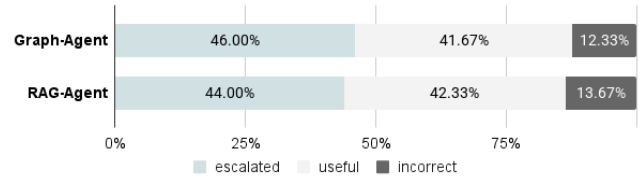

Figure 3: Model response accuracy for escalation-required inquiries.

Both Graph-Agent and RAG-Agent demonstrated competitive performance in providing generic assistance, achieving high ratios of `useful` responses at 41.67% and 42.33%, respectively[13] (Figure 3).

3.5.1 Dialogue Coherence. Tendencies for clarifying questions were similar to those for resolvable inquiries, with 42.00% of Graph-Agent dialogues featuring a clarification action type, compared to 21.33% for RAG-Agent. The number of dialogue turns (1.92 versus 1.72) and response lengths (74 versus 99) were also comparable.

3.5.2 Incorrect Responses. Graph-Agent exhibited similar tendencies for `incorrect` dialogues as with resolvable inquiries, missing critical information, but without hallucinations. On the other hand, the RAG-Agent exhibited an inclination of LLMs to provide positive responses without recognizing when they cannot resolve inquiries, highlighting the need for explicit safeguards. This resulted in RAG-Agent's tendency to go beyond its intended purposes by incorrectly claiming actions it had not actually taken during the dialogue (e.g., *"I called the card company"*, *"I verified your confirmation details"*).

4 Conclusions and Future Work

In this pilot study, we proposed a fully automated method for generating a knowledge graph from help pages, which powers a dialogue manager to resolve customer service inquiries. Our main objectives were to facilitate coherent dialogue through clarification questions to address incompleteness and to implement safeguards that improve intent identification and prevent response hallucinations.

We demonstrated that our proposal matches or outperforms a RAG baseline model for both resolvable and escalation-required inquiries while producing more coherent and natural responses.

We intend to enhance our proposal to accommodate help pages with a more flexible structure, ensuring that responses include all critical information, regardless of proximity to the corresponding problems. We also aim to validate our approach in other domains.

Acknowledgments

Presenter Biography

István Varga is a Research Engineer at Recruit Co., Ltd. Megagon Labs. He received his B.S. from Babeş-Bolyai University, Romania, and his M.S. and PhD degrees from Yamagata University, Japan. His main research interests and contributions focus on natural language processing, information extraction and recommender systems.

---

[13]A response was deemed `useful` if both annotators marked it as such; otherwise, it was considered `incorrect`. Inter-annotator agreement was moderate at 0.43.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] Janarthanan Balakrishnan and Yogesh K Dwivedi. 2024. Conversational commerce: entering the next stage of AI-powered digital assistants. *Annals of Operations Research* 333, 2 (2024), 653–687.

[3] Yang Deng, Lizi Liao, Zhonghua Zheng, Grace Hui Yang, and Tat-Seng Chua. 2024. Towards human-centered proactive conversational agents. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 807–818.

[4] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.

[5] Lihui Liu, Blaine Hill, Boxin Du, Fei Wang, and Hanghang Tong. 2023. Conversational question answering with reformulations over knowledge graph. *arXiv preprint arXiv:2312.17269* (2023).

[6] Eric WT Ngai, Maggie CM Lee, Mei Luo, Patrick SL Chan, and Tenglu Liang. 2021. An intelligent knowledge-based chatbot for customer service. *Electronic Commerce Research and Applications* 50 (2021), 101098.

[7] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. [n. d.]. Unifying Large Language Models and Knowledge Graphs: A Roadmap, 2023. *arXiv preprint arXiv:2306.08302* ([n. d.]).

[8] Shaopeng Wei, Jun Wang, Yu Zhao, Xingyan Chen, Qing Li, Fuzhen Zhuang, Ji Liu, Fuji Ren, and Gang Kou. 2022. Graph learning and its advancements on large language models: A holistic survey. *arXiv preprint arXiv:2212.08966* (2022).

[9] Guanming Xiong, Junwei Bao, and Wen Zhao. 2024. Interactive-kbqa: Multi-turn interactions for knowledge base question answering with large language models. *arXiv preprint arXiv:2402.15131* (2024).

[10] Zhentao Xu, Mark Jerome Cruz, Matthew Guevara, Tie Wang, Manasi Deshpande, Xiaofeng Wang, and Zheng Li. 2024. Retrieval-augmented generation with knowledge graphs for customer service question answering. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2905–2909.

[11] Yichi Zhang, Zhuo Chen, Lingbing Guo, Yajing Xu, Wen Zhang, and Huajun Chen. 2024. Making large language models perform better in knowledge graph completion. In *Proceedings of the 32nd ACM International Conference on Multimedia*. 233–242.

[12] Yuqi Zhu, Xiaohan Wang, Jing Chen, Shuofei Qiao, Yixin Ou, Yunzhi Yao, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2024. LLMs for knowledge graph construction and reasoning: Recent capabilities and future opportunities. *World Wide Web* 27, 5 (2024), 58.