FollowGPT: A Framework of Follow-up Question Generation for Large Language Models via Conversation Log Mining

Ziliang Zhao
Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China
zhaoziliang@ruc.edu.cn

Shiren Song
Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China
shiren.song@ruc.edu.cn

Zhicheng Dou*
Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China
dou@ruc.edu.cn

Abstract

During interactions between users and Large Language Models (LLMs), users often engage in multi-turn questioning. Understanding the user's potential follow-up intents and generating follow-up question candidates for the user is crucial for enhancing their experience with LLMs. Existing methods for follow-up question generation mainly rely on hand-crafted rules, the internal knowledge of LLMs, or the integration of external knowledge. However, these approaches fail to effectively leverage *real-world user follow-up intents* when interacting with LLMs, resulting in generated questions that do not meet the needs of practical scenarios. In this paper, we propose *FollowGPT*, a model that mines user follow-up intents from *user-LLM conversational logs*. However, directly introducing raw conversation logs leads to significant noise and sparsity issues. Therefore, to address *noises*, FollowGPT adopts a hierarchical filtering strategy for data cleaning. To mitigate the *sparsity* issue, FollowGPT employs data synthesis methods to augment the log data across three dimensions: topic diversity, intent transition diversity, and negative sample diversity. The processed data is then consolidated into a new dataset named *ShareFQG* for both training and evaluation. Finally, we train FollowGPT using a two-stage training framework involving supervised fine-tuning and preference optimization. In our experiments, we evaluate on both the ShareFQG test set and a publicly available dataset, FollowupQG, using both automated metrics and GPT-4o-based comparative evaluation. The experimental results demonstrate that our method outperforms existing baselines in various metrics, including lexical similarity, semantic similarity, and GPT-4-based evaluation for follow-up question generation, demonstrating FollowGPT's effectiveness.

CCS Concepts

• **Information systems** → **Language models**.

Keywords

Follow-up Question, Large Language Model, Conversational Search

*Zhicheng Dou is the corresponding author.

CIKM '25, November 10–14, 2025, Seoul, Republic of Korea

ACM ISBN 979-8-4007-2040-6/2025/11
https://doi.org/10.1145/3746252.3761401

ACM Reference Format:
Ziliang Zhao, Shiren Song, and Zhicheng Dou. 2025. FollowGPT: A Framework of Follow-up Question Generation for Large Language Models via Conversation Log Mining. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM '25), November 10–14, 2025, Seoul, Republic of Korea.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3746252.3761401

1 Introduction

The proliferation of Large Language Models (LLMs) represented by ChatGPT [34] and DeepSeek[22] has transformed the paradigm of information seeking [28–30, 32]. Users are shifting from traditional keyword-based matching and ranking in Web search engines to interactive conversations with LLMs [1, 12]. Despite significant advancements in human-computer interaction, current LLMs remain limited to passively responding to user questions, lacking the ability to proactively guide conversations [10]. In light of this, the **Follow-up Question Generation (FQG)** task [6, 35] has emerged as a research hotspot. FQG focuses on automatically generating more insightful follow-up question candidates aligned with user interests, based on the user's initial question and the LLM's response, either for user selection or as prompts to stimulate further exploration of related topics and knowledge. As illustrated in Figure 1, FQG builds upon the user's initial question and its answer to propose more thought-provoking questions, such as "Do these animals protect themselves from sunburn by generating corresponding pigments (e.g., darkening their skin color) internally?" This not only encourages users to ask further questions but also facilitates deeper cognitive engagement. Currently, this mechanism has been widely adopted in mainstream LLM products, including Google Assistant, Microsoft Copilot, and Baidu ERNIE Bot.

Recently, studies on FQG have made notable progress across various *domain-specific scenarios*. For instance, Gupta et al. [8] focus on the medical domain, collecting posts from Reddit to fine-tune the T5 model [38] for FQG. Rony et al. [39] improve the accuracy of follow-up questions in in-car environments with chain-of-thought prompting [47]. Differently, some studies have explored enhancing FQG in *open-domain scenarios*. For example, Kiesel et al. [15] investigate FQG using conversational search datasets. Ge et al. [6] propose a knowledge-graph-enhanced framework to improve the informativeness and coherence of follow-up questions. Liu et al. [23] introduce an "Identify-Select-Fuse" framework for generating follow-up questions. Meng et al. [27] release the first FQG dataset *FollowupQG* for information-seeking by collecting real-world data from Reddit. However, none of these methods extract follow-up intents from **real-world user interactions with LLMs**, making it challenging

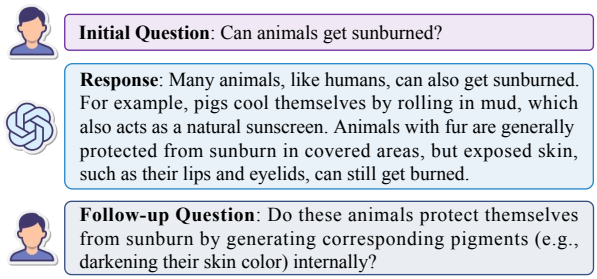

Figure 1: Examples of follow-up question generation.

to generate more realistic follow-up questions that reveal authentic user follow-up intents in open-domain user-LLM interactions.

To this end, in this paper, we propose **FollowGPT**, which **leverages human-LLM conversation logs to generate follow-up questions that align with real-world user follow-up intents** for open-domain user-LLM interfaces. We observe that directly utilizing the raw conversation logs to guide FQG presents two key challenges: **First**, the logs usually contain substantial conversational noises with low informational value, making it difficult for models to discern authentic follow-up intent. For instance, when a user first asks "What is Transformer?" followed immediately with "When did Apple release its latest iPhone?", the latter question clearly fails to constitute a meaningful follow-up to the former. **Second**, the imbalanced distribution of topics and intents across conversation logs hinders the model's ability to generate diverse and generalizable follow-up questions. For example, technical forum logs may predominantly feature discussions on AI topics while containing scarce dialogue data for other domains like healthcare or law, resulting in weaker model generalization for these underrepresented areas. The two issues hinder the utilization of logs.

FollowGPT attempts to better utilize logs from two aspects. First, to tackle log **noises**, we design a **hierarchical filtering strategy** to clean logs. Specifically, for simple noise types like extreme length, topic shifts, and surface-level repetitions, we employ text-matching rules and semantic similarity filtering. For hard noise types like keyword mismatches, logical conflicts, and excessive specialization, we design prompt templates to leverage LLMs for judgment. Besides, **sparsity** in logs still hampers model generalization across diverse scenarios. To mitigate this, we augment the data from three perspectives: (1) **Topic Diversity**: We analyze topic distributions in conversation logs using external knowledge bases (e.g., Open Directory Project, ODP), identify underrepresented or missing domains, and employ LLMs to generate synthetic question topics to improve coverage. (2) **Intent Transition Diversity**: We cluster and analyze question intent transitions in logs, categorizing them into three core types: Refinement, Comparison, and Extension, to guide diverse generation. (3) **Hard Negative Augmentation**: We use LLMs to construct various hard negative follow-up questions as preference data, enhancing the model's ability to distinguish between valid and invalid follow-up questions. Based on the filtered and synthesized data, we construct a new dataset **ShareFQG** used for training (ShareFQG-train) and evaluation (ShareFQG-test) for FQG in open-domain user-LLM interactions. Finally, we train the FollowGPT using ShareFQG-train with Supervised Fine-Tuning (SFT) followed by Direct Preference Optimization (DPO).

In our experiments, we apply the ShareFQG-test for in-domain evaluation. Additionally, we also utilize FollowupQG [27] (without training data) as a supplemental out-of-domain benchmark. To evaluate the quality of generated follow-up questions and the model's ability to capture genuine user intents, we employ both automatic evaluation metrics and GPT-4-based simulated human evaluation. Experimental results demonstrate that FollowGPT outperforms strong baselines in terms of follow-up question lexical similarity, semantic similarity, and diversity, validating the effectiveness of our methods. We further conduct an ablation study and a case study to illustrate the performance of FollowGPT intuitively.

The contributions of this paper include:

- We introduce FollowGPT, the first model that leverages real-world user-LLM conversation logs for FQG.
- Through our data filtering and synthesis pipeline, we construct ShareFQG for both training and evaluation.
- Comprehensive experiments across multiple benchmarks demonstrate FollowGPT's effectiveness.

2 Related Work

2.1 Datasets for Follow-up Question Generation

Existing FQG datasets mainly focus on some domain-specific scenarios. For example, LearningQ [3] and InquisitiveQG [17] are two educational datasets that primarily emphasize text comprehension rather than real-world user interaction patterns, making them inadequate for simulating real-world FQG scenarios. MHMC-IV [42] (academic admissions), Interview Coaching [41], and FQG [40] (job interviews) effectively capture specialized user needs but suffer from limited scalability due to small data volumes. Some datasets are derived from user interactions with search engines, including MS MARCO [33], TREC CAsT [5], Webis-Exhibition-Questions-21 [14], and Webis-Nudged-Questions-23 [7]. These datasets naturally capture sequential questioning patterns from real user interactions, aligning well with FQG objectives. Nevertheless, their primary focus on information retrieval may differ significantly from the open-domain user-LLM interaction, potentially introducing bias when directly applied to FQG training. Recently, FollowupQG [27] is proposed based on Reddit forum interactions through Web crawling and rigorous filtering. The data format in FollowupQG closely resembles real-world user-LLM interactions, but its limited data diversity makes it challenging to apply in LLM products.

To address the prevalent challenges of noise, data sparsity, and domain bias in existing datasets, we introduce a novel large-scale, high-quality dataset, ShareFQG, based on user-LLM conversation logs [44, 53, 56], which have been widely recognized as a crucial resource for uncovering user intents [43].

2.2 Methods for Follow-up Question Generation

FQG plays a crucial role in human-LLM online interactions. Existing studies mainly focus on three series of methods: structured modeling, knowledge-enhanced generation, and user feedback-based methods, aiming to improve follow-up questions from different dimensions like relevance, coherence, and diversity. Early studies employ **structured modeling** and hierarchical generation strategies. For example, Yang et al. [49] utilize neural matching models for question retrieval, and Liu et al. [25] propose HierLLM for hierarchical

question recommendation. These methods, however, demonstrate limited capability in capturing complex intents during interactions. Some studies integrate **external knowledge** to enhance question depth and informativeness. Rony et al. [39] implement chain-of-thought prompting [47] and retrieval-augmented generation for in-car conversational search. Liu et al. [23] develop EK-FQG, combining knowledge graphs with LLMs, while KG-FQG [6] introduces a two-stage framework with knowledge selection. Though effective, these methods suffer from high computational complexity and dependency on knowledge base completeness. Recently, some methods have been trying to leverage **real-world user feedback** to simulate follow-up questioning behavior. Kiesel et al. [15] utilize conversational search logs, while Gupta et al. [8] create process-based frameworks for mental health applications. FollowupQG [27] tries to mine human-human interactions to build follow-up question data. Despite capturing real-world user intent, these methods still struggle with data sparsity and noise. Moreover, these methods are only trained on relatively small-scale and domain-specific datasets, which limits their generalizability to follow-up question generation in open-domain user-LLM interactions.

2.3 Search Clarification

While FQG assumes clear user intent and focuses on anticipated interest shifts, real-world user questions often contain ambiguity requiring **clarification**. For instance, resolving "Recommend some restaurants" necessitates generating disambiguation questions like "Where is your location?" or "Which kind of restaurant do you like?". Crucially, clarification occurs before answering the original question, whereas follow-up questions extend resolved intents. This fundamental distinction separates the two tasks despite superficial similarities. Current clarification research briefly bifurcates into search and QA scenarios. Zamani et al. [50] establish foundational template-based methods using search logs, later extended through clarification panels [45]. Zhao et al. [54, 55] enhance this by incorporating Web search results to better capture user intent. With LLM advancements, in QA clarification, Kim et al. [16] employ retrieval-augmented models for ambiguity resolution, while Lee et al. [19] develop an InstructGPT-powered framework encompassing ambiguity detection, clarification generation, and response integration. In contrast, FQG operates within established conversation contexts, utilizing prior questions and system responses to drive coherent, contextually grounded inquiries that deepen user engagement rather than resolve ambiguities.

3 FollowGPT

3.1 Problem Formulation

In multi-turn user-LLM interactions, the FQG task aims to generate potential follow-up questions based on the user's previous question and the system's response, then **display the generated questions for the user to select**, thereby guiding users to further explore relevant information. Existing studies typically adopt a simplified modeling approach, where follow-up questions are generated solely based on the semantic information of the user's initial question and the system's response [27]. This formal definition focuses on single-turn follow-up generation, avoiding the interference of complex historical contexts while avoiding the multi-turn sparsity issue.

In this paper, we follow this **single-turn modeling approach**. Formally, the input is defined as a tuple (q, r), where:

- $q \in \mathcal{Q}$ represents the user's initial question, belonging to the user's question space $\mathcal{Q}$,
- $r \in \mathcal{R}$ denotes the system's response to q, belonging to the system's answer space $\mathcal{R}$.

The objective of FQG is to learn a function $f : \mathcal{Q} \times \mathcal{R} \rightarrow \mathcal{Q}$ that maps the input pair (q, r) to a potential follow-up question q', i.e., $q' = f(q, r)$, which is intended to guide users toward deeper exploration of information related to the initial question while maintaining semantic coherence with the preceding answer.

It is worth noting that although Balog [2] has identified several factors influencing users' follow-up questions, including personal interests, existing knowledge, and understanding of system capabilities, we simplify the problem by modeling a "generic user" without personalized considerations. Thus, the core challenge lies in extracting key information from (q, r) while accounting for potential shifts in user intent, thereby generating q' that aligns with the user's next-step questioning intent, without explicitly addressing individual differences among users.

3.2 Method Framework

In this paper, we propose the **FollowGPT** framework for generating user follow-up question candidates based on real-world user-LLM conversation logs. As illustrated in Figure 2, our methodological framework consists of two main components: (1) the data construction phase, including data extraction, filtering, and synthesis, and (2) the two-stage model training, comprising supervised fine-tuning and direct preference optimization.

First, to address noises from invalid follow-up questions in conversation logs, we first extract (q_1, r_1, q_2) triples from user-LLM interaction logs. We then design a **hierarchical filtering strategy** to clean the raw data. For simple noise samples (e.g., extreme length, topic drift, surface-level repetition), we employ text matching rules and semantic similarity filtering. For hard noise samples (e.g., keyword mismatch, logical conflicts, excessive specialization), we utilize strong LLM for judgment, with carefully designed prompt templates to ensure consistent discrimination. This process effectively removes irrelevant data and keeps useful follow-up intents.

Furthermore, to mitigate data sparsity in conversation logs, we employ **data synthesis** to expand the dataset from three perspectives: topic diversity, intent transition diversity, and negative sample diversity. First, based on the Open Directory Project (ODP)[1], we analyze topic distributions in conversation logs to identify underrepresented or missing domains, then use LLMs to generate corresponding topic-specific questions and answers, thereby improving data coverage. Second, we perform cluster analysis on question intent transition patterns in logs and categorize them into three core intent types: vertical refinement, horizontal comparison, and expansive extension. We then design corresponding prompts to guide LLMs in generating follow-up questions with different intents, ensuring the dataset covers a broader range of user needs. Finally, we also generate negative samples for preference optimization. Specifically, we employ LLMs to construct diverse invalid follow-up questions covering redundancy, ambiguity, logical errors,

[1] http://odp.org: An open-source collection of Web topics

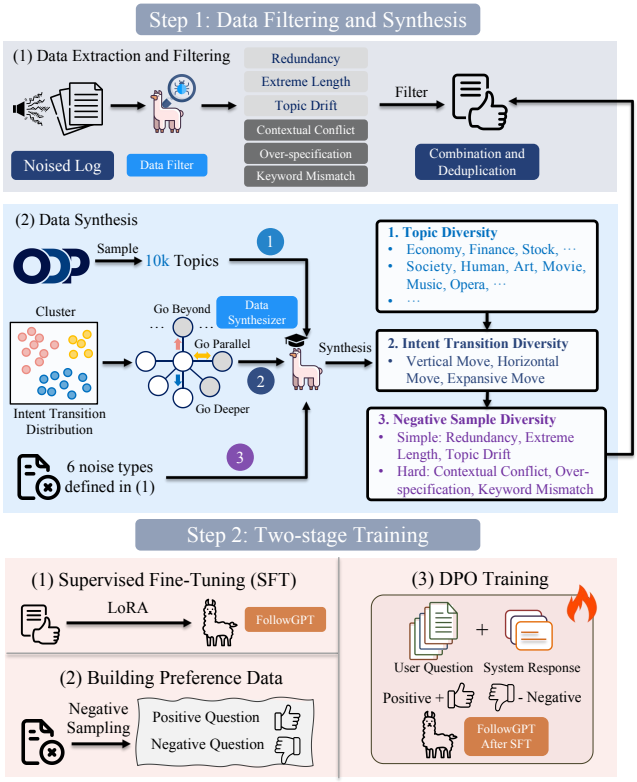

Figure 2: The Overall Framework of FollowGPT.

etc., enhancing the model's ability to distinguish between valid and invalid follow-up questions. After the data filtering and synthesis process, we build a new dataset **ShareFQG** for the training and evaluation of open-domain FQG in user-LLM interaction.

In the **model training** phase, we adopt a two-stage approach: The first stage involves Supervised Fine-Tuning (SFT) on high-quality datasets to help the model initially master the FQG task and learn generation patterns for different intent categories. The second stage employs Direct Preference Optimization (DPO) using constructed preference data to refine the model, making it more likely to generate user-preferred follow-up questions while avoiding redundant or irrelevant ones. By directly optimizing the probability ratio between positive and negative samples, DPO further improves both question quality and user acceptance.

3.3 Conversation Log Pre-processing

In user-LLM interactions, a log session are recorded as a (question-response) pair: $\mathcal{R} = \{(q_1, r_1), (q_2, r_2), \ldots, (q_k, r_k)\}$, where q_i represents the user's question in the i-th turn and r_i denotes the corresponding system response.

Direct extraction of (q_{t-1}, r_{t-1}, q_t) triples from raw logs often yields the questions q_{t-1} and q_t with strong contextual dependencies (e.g., "Where is it then?"), making them semantically incomplete in isolation. To address this, we employ **question reformulation** technology to transform context-dependent information into self-contained, context-independent sentences while standardizing question expressions to reduce noise caused by linguistic variations. This strategy has been widely adopted in dialogue systems and information retrieval tasks with demonstrated effectiveness [31].

Specifically, following the design of previous generative question reformulation studies [31], we design the prompt template for question reformulation shown in Figure 3 (P1). In the prompt, a demonstration is selected for in-context learning. Through this template, each original question in the logs is converted into an independent sentence, eliminating contextual dependencies. We then extract all (q_{t-1}, r_{t-1}, q_t) as (q, r, q') triples from processed logs as the foundational dataset $\mathcal{D}$.

3.4 Conversation Log Noise Filtering

Real-world user interaction logs with LLMs often contain substantial noise. Adjacent question pairs (q_{t-1}, q_t) in multi-turn conversations may be invalid follow-up intent relationships. For example, when q_{t-1} is "What is deep learning?" and q_t is "Write an AI program.". Although the two are related, the latter shifts the topic from the former and does not constitute a good follow-up intent.

Therefore, directly treating all consecutive question pairs as valid follow-ups would introduce noise samples misaligned with actual needs. This section details our approach to filtering raw log data, ensuring reliability and relevance for FQG training.

3.4.1 Noise Type Analysis. It is challenging to directly define "what is a good follow-up question". Instead, we investigate the opposite question: "What makes a follow-up question ineffective". To this end, by analyzing a large amount of (q_{t-1}, q_t) in the logs, we derive a noise taxonomy shown in Table 1 where the example initial question q_{t-1} is "*In contrastive learning, do we always need labeled data for training?*". This taxonomy supports subsequent data filtering and model optimization processes.

3.4.2 Filtering Pipeline. We implement a two-tiered filtering approach to filter both simple and hard noises:

Simple Noises Filtering. For the simple noisy types "extreme length", "redundancy", and "topic drift", first, since extreme long questions are not suitable as follow-up questions [27], we only retain questions whose length falls within the range of 5 to 32 words. After that, for redundancy and topic drift, we apply the **semantic filtering** by computing cosine similarity between (q_{t-1}, r_{t-1}) and q_t embeddings via a language encoder E (like BERT):

$$s = \cos\left(E(q_{t-1}, r_{t-1}), E(q_t)\right). \tag{1}$$

Referring to the construction of MS-MARCO conversational search data [33] that applies threshold and semantic similarity to filter data, we set two thresholds τ_l and τ_h. When $s < \tau_l$, we consider the relevance between q_t and q_{t-1} to be very low, thus qualifying as topic drift, that is, an abrupt shift to another subject. When $s > \tau_h$, we consider the relevance between q_t and q_{t-1} to be too high, likely indicating repetitive content, and thus it should also be filtered. Finally, only data with $\tau_l \le s \le \tau_h$ are retained. In this paper, we set τ_l to be 0.5 and τ_h to be 0.9.

Hard Noise Filtering. It is challenging to filter the hard noises in the logs using rule-based methods. Therefore, we rely on a powerful LLM with unified prompt templates shown in Figure 3 (P2) to discover three kinds of hard noisy samples:

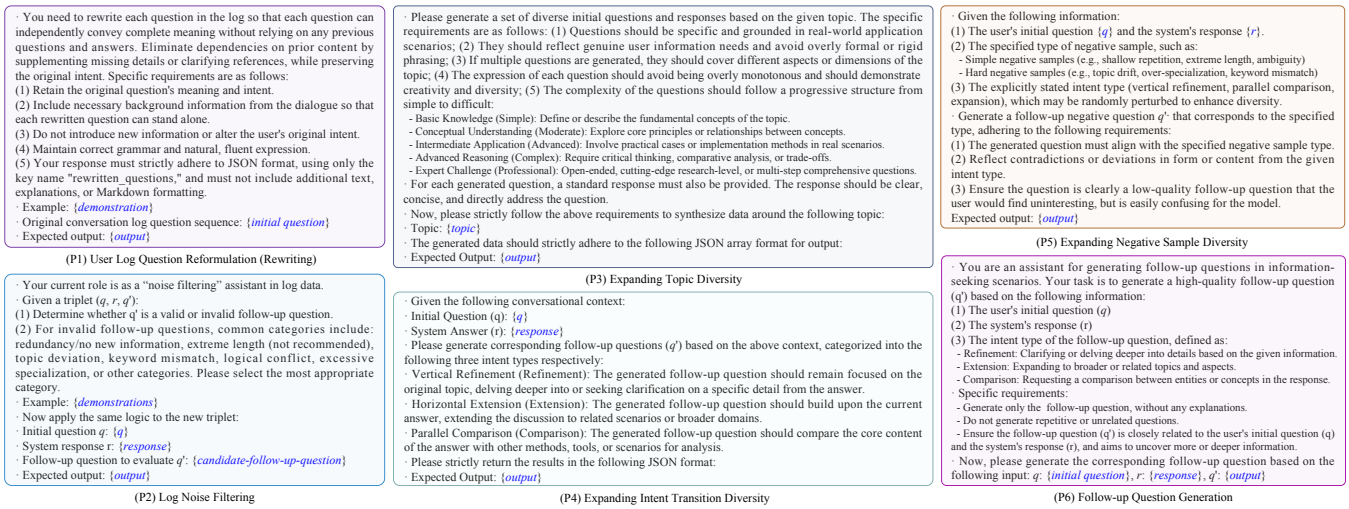

Figure 3: All Prompts used including data processing, filtering, synthesis, and follow-up question generation.

Table 1: Noisy sample categories and examples mined from conversation logs

Noise Type	Category	Description	Example Follow-up q_t
Simple	Redundancy	Follow-ups are highly repetitive or superficially rephrased without substantive new content.	"So, do we need labeled data for contrastive learning or can we proceed without any labels?"
	Extreme Length	Overly short (unclear intent) or long (redundant) questions reduce effectiveness.	"Really?", "Could you provide all hyperparameters and logging details for large-scale multi-GPU..."
	Topic Drift	Follow-ups clearly disconnected from the current dialogue topic.	"How can I make yeast-fermented bread at home?"
Hard	Keyword Mismatch	Surface-level keyword matches (e.g., Transformer) with semantic disconnects.	"I want to set up a transformer to boost voltage from 110V to 220V - how?"
	Contextual Conflict	Logical contradictions with preceding responses.	"So this proves contrastive learning can never work without labels, right?"
	Over-Specialization	Over-reliance on niche scenarios limiting generalizability.	"For our internal vision pipeline, if managers insist on specific labels, can we still use contrastive learning?"

- **Keyword Mismatch:** Disambiguating keyword intent versus dialogue context.
- **Contextual Conflict:** Identifying logical inconsistencies with prior responses.
- **Over-Specialization:** Detecting non-generalizable scenario-specific queries.

Our multi-stage filtering processes 22,109 initial (q_{t-1}, r_{t-1}, q_t) triples sampled from real user-LLM logs, sequentially applying simple then hard negative filters. The final data after filtering contains 13,680 high-quality samples (q, r, q').

3.5 Data Synthesis and Diversity Enhancement

The noise filtering stage ensures the quality of follow-up question data mined from conversation logs. However, relying solely on these data for training would lead to the sparsity issue, that is, overfitting to the obtained data distribution, thereby limiting the depth and breadth of FQG. To this end, we propose a multi-dimensional data synthesis approach. Specifically, we introduce external knowledge and LLM-based synthesis strategies to expand and balance data from three key dimensions: **topic diversity**, **intent transition diversity**, and **negative sample diversity**, systematically constructing diverse data samples to enhance the model's ability to capture implicit patterns in follow-up questions.

3.5.1 Topic Diversity. The analysis of real-world conversation logs reveals an imbalanced distribution of topics in users' initial questions q [4], usually leading to insufficient learning of low-frequency topics. To address this limitation, we first establish a comprehensive hierarchical topic system based on the Open Directory Project (ODP), then map each q in the logs to one existing category to identify underrepresented topics. For these sparse or missing topics, we employ a powerful LLM to generate multiple (q, r) pairs based on given keywords and topic labels, thereby expanding the corpus with diverse and representative "initial question-answer" combinations. This process not only compensates for the natural sparsity of log data but also provides broader topic coverage for FQG. The specific prompt template is shown in Figure 3 (P3).

3.5.2 Intent Transition Diversity. Beyond the topics of initial questions q, the intent transitions from (q, r) to q' also reflect the depth and breadth of conversations. While some obvious transition types (e.g., vertical refinement or horizontal comparison) can be identified through manual observation, relying solely on empirical induction may overlook other subtle yet important patterns. Therefore, we

adopt a data-driven approach: First, we apply a powerful LLM to identify potential intent transition phrases in ShareGPT logs. After that, we apply a sentence embedding model to generate semantic vector representations for each phrase, which are then clustered to form phrase groups. Specifically, we use the Ward method to calculate inter-cluster similarity and automatically categorize these transition phrases in high-dimensional space. Finally, we further apply a powerful LLM to generate clear, interpretable labels for each cluster to systematically identify all possible intent transition patterns. Combining mined labels, we summarize three core intent transition types in follow-up questions: (1) **Vertical Refinement**: Delving deeper into details of the original question or the LLM response. (2) **Horizontal Comparison**: Contrasting different perspectives or options. (3) **Expansive Extension**: Introducing new related dimensions or perspectives based on the current topic. This approach ensures completeness in intent transition types while improving overall data quality and generalization. For each intent type, we further apply a powerful LLM to generate new (q, r, q') triples, balancing and enriching the diversity of intent transitions in the training set. The prompt template is shown in Figure 3 (P4).

3.5.3 Negative Sample Diversity. During the noise filtering stage mentioned in Section 3.4, we have identified various low-quality follow-up question types (e.g., extreme length, topic drift, contextual conflicts). To help the model distinguish between genuine follow-up intents and noisy follow-up intents, we further employ an LLM to synthesize diverse q'^- samples: On one hand, we provide fine-grained instructions based on noise type labels to generate corresponding invalid questions. On the other hand, we explicitly introduce random perturbations to intent types like "vertical refinement", "horizontal comparison", or "expansive extension", ensuring varied manifestations even within the same noise category. The resulting (q, r, q'^-) samples provide richer training signals for the model's ability to discriminate negative samples and understand conversations. The prompt template is shown in Figure 3 (P5).

3.6 Two-stage Training

The training process of FollowGPT consists of two key stages: Supervised Fine-Tuning (SFT) and Direct Preference Optimization (DPO). The SFT stage enables the model to comprehend various follow-up question intents and acquire fundamental generation capabilities, while the DPO stage further aligns the generated outcomes with human preferences, thereby enhancing the quality of the model's question generation. This two-stage approach ensures that FollowGPT not only learns to generate contextually appropriate follow-up questions but also refines its outputs to better match users' actual information needs and preferences. The SFT stage establishes the model's baseline performance through exposure to high-quality training examples, while the DPO stage fine-tunes this capability by incorporating synthesized preference signals to distinguish between more and less desirable responses.

3.6.1 Supervised Fine-Tuning (SFT). The SFT stage involves direct training of the FQG model using high-quality training data. In this paper, through noise filtering and data synthesis, we construct two datasets: a real-user log dataset $D_{\text{log}} = \{(q, r, q'^+, q'^-)\}$ and a synthetic dataset $D_{\text{syn}} = \{(q, r, q'^+, q'^-)\}$, where q represents the user's initial question, r the corresponding system response, and q'^+ and q'^- denote high-quality positive and negative follow-up question samples, respectively. Since only positive samples are required for the SFT stage, we extract and merge all positive samples from these datasets to form the final training dataset for SFT:

$$D_{\text{SFT}} = \{(q, r, q'^+) \mid (q, r, q'^+, q'^-) \in (D_{\text{log}} \cup D_{\text{syn}})\}, \quad (2)$$

while the corresponding high-quality negative samples q'^- will be used in the next stage DPO for preference optimization. During training, we carefully design a prompt template T to guide the model in following instructions and generating high-quality follow-up questions, as shown in Figure 3 (P6). The objective of SFT training is to maximize the log-likelihood of positive samples q'^+:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\sum_{(q,r,q'^+) \in \mathcal{D}_{\text{SFT}}} \log p(q'^+ | T(q, r); \theta). \quad (3)$$

3.6.2 Direct Preference Optimization (DPO). While SFT enables the model to generate follow-up questions for different intents, it may still produce questions that do not align with user preferences. To further improve the alignment between generated follow-up questions and human preferences, we employ DPO [37]. Specifically, we use constructed high-quality positive and negative samples to form a dataset with explicit preference relationships. The sample format for DPO training is (q, r, q'^+, q'^-), where the model learns to prefer generating q'^+ over q'^- based on preference signals. The objective of DPO training is:

$$\mathcal{L}_{\text{DPO}}(\theta) = -\sum_{(q,r,q'^+,q'^-) \in \mathcal{D}_{\text{DPO}}} \log \sigma \left(\beta \log \frac{p(q'^+ | q, r; \theta)}{p(q'^- | q, r; \theta)} \right), \quad (4)$$

where $\sigma(\cdot)$ is the sigmoid function, and β is a temperature hyperparameter that controls the strength of preference comparison. Unlike traditional reinforcement learning, DPO does not require explicit reward function estimation. Instead, it directly optimizes the probability between positive and negative samples, reducing the generation probability of hard negative samples and thereby improving the quality of FQG.

4 Experiments

4.1 Log and Data

All data are built based on the ShareGPT[2] user-LLM log. We apply the collected dataset **ShareFQG-train** for SFT and DPO training of FollowGPT, and apply **ShareFQG-test** for evaluation[3]. We further use **FollowupQG**, a public FQG dataset widely used for performance evaluation [23, 26]. It contains over 3,000 (q, r, q') triplets collected from the Reddit forum "Explain Like I'm Five" (ELI5), where users provide simplified explanations for open-ended questions. Unlike other datasets, FollowupQG features questions with richer information needs, answers from real user replies, and follow-up questions demonstrating advanced cognitive skills like application and association. We use this dataset as an out-of-domain benchmark to evaluate FollowGPT's generalization. Detailed statistics of both evaluation datasets are shown in Table 2.

[2]https://huggingface.co/datasets/anon8231489123/ShareGPT_Vicuna_unfiltered
[3]ShareFQG: https://huggingface.co/datasets/ZillionZhao/ShareFQG

Table 2: Statistics of two FQG Datasets

Metric	ShareFQG	FollowupQG
Training Set Size	13,680 (27,360)	2,790 (5,580)
Validation Set Size	1,521	500
Test Set Size	1,521	501
Avg. Question Length	79.5	43.6
Avg. Answer Length	234.3	143.5

4.2 Evaluation Metrics

We employ two categories of automated metrics to comprehensively evaluate generated follow-up questions: *generation quality* and *diversity*. To evaluate the generation quality, we apply BLEU [36] and ROUGE [21] to measure the lexical similarity between the generated follow-up questions and ground-truth questions, and then apply BERTScore [52] to compute semantic similarity using BERT embeddings, capturing fine-grained semantic alignment. Besides the question quality, we also evaluate the diversity of the generated follow-up questions. Specifically, we apply Distinct [24] to measure the ratio of unique unigrams/bigrams in generated texts, and TTR (Type-Token Ratio) [13] to compute the ratio of unique words to total words, reflecting vocabulary richness.

4.3 Baseline Models

We compare FollowGPT with baselines in four categories:

- **Small-scale PLMs**: Sequence-to-Sequence models including BART [20] and T5 [38], fine-tuned on FQG datasets.
- **In-Context Learning LLMs**: Open-source LLMs (Mistral-7B [11], Qwen2.5-7B [48], ChatGLM4-9B [51]) using few-shot prompting without fine-tuning.
- **Knowledge-Enhanced**: EK-FQG [23], a state-of-the-art method combining GPT-3.5-turbo with knowledge graphs through recognition, selection, and fusion stages.
- **Supervised Fine-Tuning**: Qwen2.5-3B [48] fine-tuned with high-quality annotated data (without DPO).

4.4 Experimental Details

The FollowGPT model implementation is based on the LLaMA-Factory framework [57], with inference acceleration provided by vLLM [18]. All training processes are conducted on a single NVIDIA A800-SXM4-80GB GPU, using PyTorch 2.4.0 and Transformers 4.45.2. During the SFT stage, we conduct efficient parameter adaptation through low-rank adaptation (LoRA) [9]. Specifically, we set rank=8, initial learning rate=$1e-4$, batch size=8, with gradient accumulation steps=4, cosine annealing learning rate scheduler, and training epochs=5. In the DPO stage, we set the learning rate to $1e-5$, the scaling factor β=0.1, and adjust the batch size to 16.

The FollowGPT framework utilizes LLaMA3-3B as its foundational model. Throughout this study, all instances described as "powerful LLMs" refer exclusively to invocations of the official GPT-4o API. In the intent transition diversity enhancement module, the BGE-m3 embedding model[4] serves as our semantic representation encoder. Additionally, during the construction of the ShareFQG, we configure different temperatures for different tasks: For log data pre-processing, the temperature coefficient of question rephrasing prompt templates is set to 0.2 to ensure textual coherence and stability. During noise filtering and negative sample generation, the temperature coefficient is fixed at 0.0 to guarantee deterministic text output. For data synthesis and diversity enhancement, we use a temperature coefficient of 1.0 to stimulate the model's creativity and improve synthetic data diversity. The top-p parameter is consistently set to 1.0 throughout all processes.

[4]https://huggingface.co/BAAI/bge-m3

4.5 Main Experimental Results

We conduct evaluations on the ShareFQG and FollowupQG datasets, with the results presented in Table 3. Bold values in the tables indicate the best performance, while underlined values denote the second-best. Among these metrics, FollowGPT demonstrates significant advantages in text quality for generating follow-up questions while maintaining comparable performance in text diversity.

Specifically, first, in terms of generation quality, FollowGPT achieves state-of-the-art (SOTA) results on ShareFQG, demonstrating its success in learning users' follow-up intents from user-LLM conversation logs and generating follow-up questions that better align with user preferences. Additionally, despite not having been exposed to the FollowupQG dataset, FollowGPT still outperforms other models on FollowupQG, indicating its strong generalization and the absence of overfitting on our ShareFQG dataset. Regarding the diversity of generation results, FollowGPT does not exhibit particularly high metrics, which may stem from the inherent trade-off between generation quality and diversity: supervised fine-tuned models improve accuracy by fitting training data but may sacrifice output diversity, whereas in-context learning models rely on example-guided generation, which can produce more varied expressions but may not ensure that the follow-up questions better match user preferences. However, it is worth noting that while FollowGPT significantly outperforms others in quality, its Dist-2 scores on ShareFQG (62.55) and FollowupQG (70.65) are both close to the optimal levels, indicating that the diversity of follow-up questions generated by FollowGPT remains comparable.

Compared with FollowGPT, PLMs exhibit lower performance in both quality and diversity due to their limited parameter size. For example, BART and T5 achieve BLEU-1 scores of 28.88 and 23.87, respectively, on the ShareFQG dataset, significantly lower than the proposed FollowGPT. Besides, for in-context learning LLMs, although they achieve higher diversity metrics (e.g., Mistral-7B's Dist-2 reaches 63.98 in ShareFQG), their quality metrics still fall short of FollowGPT. EK-FQG [23] leverages knowledge graphs to enhance generation capabilities, achieving competitive diversity metrics (e.g., Dist-1 of 31.19). However, its text quality metrics are significantly inferior to FollowGPT, suggesting that while the generated questions are abundant in diversity, they may not fully align with users' real follow-up intents. Finally, although the Qwen2.5-3B model undergoes the SFT stage, its quality metrics (e.g., BLEU, ROUGE, and BERTScore) are slightly lower than FollowGPT, likely due to the lack of data synthesis and DPO optimization.

4.6 Ablation Study

To evaluate the effectiveness of each module, we conduct the following ablation experiments on both the ShareFQG and FollowupQG

Table 3: Main experimental results of the proposed method on ShareFQG and FollowupQG datasets.

Dataset	Category	Model	Generation Quality					Generation Diversity		
			BLEU-1	BLEU-4	ROUGE-1	ROUGE-L	BERTScore	Dist-1	Dist-2	TTR
ShareFQG	Small-scale Pretrained Models	BART	28.88	8.41	36.24	27.21	60.85	20.79	60.43	19.97
		T5	23.87	5.74	30.58	23.80	58.84	21.18	60.05	20.17
	In-Context Learning	Mistral-7B	23.42	5.42	31.39	23.28	61.13	**22.12**	**63.98**	21.43
		Qwen2.5-7B	21.05	5.22	29.56	23.02	56.64	21.35	60.80	20.89
		ChatGLM4-9B	22.58	5.44	30.85	23.54	59.53	21.47	62.57	20.97
	Knowledge-Enhanced	EK-FQG†	19.68	4.81	28.76	22.55	58.67	22.07	61.40	**21.60**
	Supervised Fine-Tuning	Qwen2.5-3B	30.43	8.98	37.62	29.08	64.34	20.21	59.97	19.56
	Our Method	FollowGPT	**33.04**	**11.73**	**40.02**	**31.96**	**67.03**	20.59	62.55	19.85
FollowupQG	Small-scale Pretrained Models	BART	8.00	1.45	15.46	11.68	38.75	26.78	68.89	25.97
		T5	6.90	1.20	14.28	11.33	33.86	26.93	64.47	26.24
	In-Context Learning	Mistral-7B	8.42	1.17	13.99	10.32	40.26	29.75	**70.86**	29.41
		Qwen2.5-7B	7.60	1.19	13.69	10.59	39.07	29.15	69.48	**30.87**
		ChatGLM4-9B	8.40	1.22	14.52	10.84	39.82	30.04	69.65	29.82
	Knowledge-Enhanced	EK-FQG†	7.07	1.09	12.67	9.73	38.16	**31.19**	68.82	28.98
	Supervised Fine-Tuning	Qwen2.5-3B	9.10	1.21	15.59	11.40	37.42	21.03	64.61	20.45
	Our Method	FollowGPT	**9.64**	**1.90**	**16.12**	**12.79**	**43.28**	28.61	70.65	28.04

Table 4: Ablation study results of the proposed method on the ShareFQG dataset.

Ablation Setting	BLEU-1	BLEU-4	ROUGE-1	ROUGE-L	BERTScore
FollowGPT	**33.04**	**11.73**	**40.02**	**31.96**	**67.03**
- Synthetic Data	26.53	6.67	33.41	25.01	59.02
- DPO	30.44	9.26	38.76	29.10	65.44
- System Responses	29.00	7.93	35.85	27.65	62.06

Table 5: Ablation study results of the proposed method on the FollowupQG dataset.

Ablation Setting	BLEU-1	BLEU-4	ROUGE-1	ROUGE-L	BERTScore
FollowGPT	9.64	**1.90**	16.12	**12.79**	**43.28**
- Synthetic Data	8.84	1.20	15.26	10.84	38.57
- DPO	9.06	1.70	15.94	11.50	41.98
- System Responses	**9.73**	1.24	**16.54**	11.80	36.84

datasets: (1) **FollowGPT w/o Synthetic Data**: Removing the synthetic data from training data and using only the original training dataset for SFT and DPO. (2) **FollowGPT w/o DPO**: Removing the DPO stage and performing only single-stage SFT training. (3) **FollowGPT w/o System Responses**: Excluding the system responses from the training data and considering only the initial user question to validate the role of responses in FQG. The experimental results are shown in Tables 4 and 5, where bold values indicate the best performance and underlined values denote the second-best.

The results in Tables 4 and 5 clearly demonstrate the effectiveness of each component in the FollowGPT model. It can be observed from the tables that, first, synthetic data is crucial. The experimental results indicate a significant decline in the generation of follow-up questions after removing data synthesis, compared to DPO and system response. This demonstrates that the synthesized data effectively mitigates the data sparsity issue in user-LLM conversation logs. Second, in the absence of DPO training, all evaluation metrics exhibit varying degrees of decline. This suggests that follow-up questions implicitly encode user preference information, thereby validating the utility of preference learning in enhancing model performance. Finally, system responses also play a critical role. Since follow-up questions are often conditioned on the system's prior outputs, excluding them during training prevents the model from effectively capturing potential shifts in user intent. The results validate the design choices of FollowGPT, demonstrating that components including synthetic data, system responses, and DPO collectively contribute to high-quality FQG.

4.7 Simulated Human Evaluation with GPT-4o

Following the experimental design of prior studies [46], we employ GPT-4o to conduct pairwise comparative evaluations between FollowGPT and baseline models, further validating the effectiveness of FollowGPT. To simulate human evaluation preferences, we adopt the setup from [27] and define three key evaluation criteria: (1) **User intent consistency**: Assess which candidate question better aligns with the original user intent. (2) **Contextual relevance**: Evaluate which candidate question maintains stronger logical coherence with the initial query and its response. (3) **Natural fluency**: Determine which candidate question exhibits more natural, fluent, and clearly articulated phrasing. For the ShareFQG dataset, due to evaluation costs, we randomly sample 500 instances from the test set for assessment. For the FollowupQG dataset, we evaluate the full test set. The prompt setting stays consistent with [46].

As shown in Figure 4, FollowGPT outperforms baseline models on both the ShareFQG and FollowupQG datasets. The win-tie-loss comparison reveals the following conclusions. First, on both datasets, FollowGPT significantly surpasses smaller pre-trained language models like BART and T5. This is because BART and T5, constrained by their parameter scales, often generate follow-up questions with incoherence or poor contextual relevance, making them less favored by users. Second, FollowGPT enables Llama3-3B to outperform larger-scale models, including Mistral-7B, Qwen2.5-7B, and ChatGLM-9B. For Qwen2.5-3B, which is also fine-tuned with supervised learning, the lack of data synthesis and DPO optimization results in weaker performance in capturing fine-grained

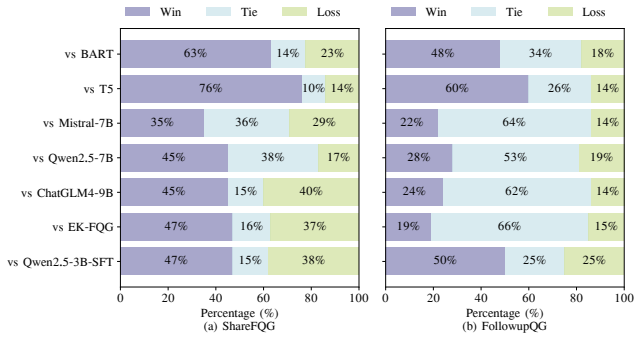

Figure 4: Model comparison results based on GPT-4o automated evaluation (Win-Tie-Loss).

user intent preferences compared to FollowGPT. Finally, since the FollowupQG dataset is collected from ELI5, where initial questions and dialogues originate from human-human interactions, the follow-up questions tend to be more divergent. This leads to a higher number of ties (where GPT-4o cannot determine a preference) on this dataset. In contrast, the ShareFQG dataset undergoes data cleaning and noise filtering, making the test set more representative with clearer intent, thus resulting in fewer ties.

4.8 Case Study

To further demonstrate the effectiveness of our method, this section selects an example from the ShareFQG dataset for analysis and explanation. Due to space limitations and readability, we choose one representative model from each baseline category, including BART, Mistral-7B, EK-FQG, and Qwen2.5-3B-SFT.

As shown in Table 6, for FQG in a middle-distance orienteering scenario, the models exhibit significant differences in the three evaluation dimensions: user intent consistency, contextual relevance, and natural fluency. First, BART, Mistral-7B, and EK-FQG fail to focus on the core strategy of "handrails", leading to follow-up questions that deviate from the key points highlighted in the system answer. While Qwen2.5-3B-SFT mentions "handrails", it lacks deeper exploration of athlete skills or navigation abilities, resulting in relatively neutral phrasing. In contrast, FollowGPT not only tightly centers on the "handrails" theme but also incorporates additional elements like "terrain difficulty" and "personal navigation skills", making the generated question more aligned with user intent and situational needs. It also demonstrates superior performance in language naturalness and content richness. This case illustrates that our method better captures critical information and aligns with user intent, thereby generating higher-quality follow-up questions.

5 Conclusion

We present FollowGPT, a novel framework for generating follow-up questions that align with real-world user intents by leveraging authentic user intent from user-LLM conversational logs. To address the challenges of noise and sparsity in raw logs, FollowGPT employs a hierarchical filtering strategy for data cleaning and introduces a multi-dimensional data synthesis approach to enhance topic diversity, intent transition diversity, and negative sample diversity. The resulting dataset, ShareFQG, serves as a comprehensive resource

Table 6: Case study of different FQG models.

Case ID: 10814

Initial Question (Q1): In a middle-distance orienteering race, what strategies can athletes use to optimize their route choice and pace?

System Answer (R): To optimize their route choice and pace in a middle-distance orienteering race, athletes can use strategies such as reading the map ahead (contextual relevance +), breaking the course into smaller sections (fluency +), and using handrails like trails or streams (intent consistency +). They should also consider physical condition and adjust pace, factoring terrain difficulty and navigation complexity.

Follow-up Question Candidates

Reference (Human): How do athletes determine the most effective handrails to use in a middle-distance orienteering race, considering factors such as terrain and navigation complexity? (intent consistency +, contextual relevance +)

BART: How can athletes specifically use the map ahead to anticipate upcoming challenges **(missing 'handrails')** (intent consistency -) and adjust pace accordingly in a middle-distance orienteering race?

Mistral-7B: What specific techniques can athletes use to effectively break down a middle-distance course **(missing 'handrails')** (intent consistency -) for optimal route planning?

EK-FQG: How can athletes effectively adjust their pace in a middle-distance orienteering race based on varying terrain difficulty? **(missing 'handrails')** (intent consistency -)

Qwen2.5-3B-SFT: How can athletes effectively use handrails (trails/streams) to guide their route choice in a middle-distance orienteering race? (intent consistency +, contextual relevance +) **(lacks elaboration on personal details)** (fluency: moderate)

FollowGPT (Our): How do athletes determine the optimal handrails (intent consistency ++ , contextual relevance ++) to use in a middle-distance orienteering race, considering factors like terrain difficulty and personal navigation skills? (fluency ++)

for training and evaluation. Through a two-stage training framework combining SFT and DPO, FollowGPT demonstrates superior performance over existing methods, as evidenced by experiments on both ShareFQG and the public FollowupQG dataset. Evaluations using automatic metrics and GPT-4o-based human simulation confirm that FollowGPT excels in lexical similarity, semantic relevance, and diversity, validating its effectiveness in practical scenarios. Our work not only advances the state-of-the-art in follow-up question generation but also highlights the importance of grounding such systems in real-world user interactions to better meet their needs. Future directions include extending the FollowGPT framework to multi-modal settings and further refining intent modeling for even more nuanced question generation.

Acknowledgments

This work was supported by National Natural Science Foundation of China No. 62272467, Beijing Natural Science Foundation No. L233008, Beijing Municipal Science and Technology Project No. Z231100010323009, and National Science and Technology Major Project No. 2022ZD0120103. The work was partially done at the Engineering Research Center of Next-Generation Intelligent Search and Recommendation, MOE.

GenAI Usage Disclosure

In our methods and experiments, GenAI is employed for data generation and comparative evaluation, which is totally declared in this paper. We further apply GenAI to generate complicated Table structures (Table 3) followed by manual editing. In the writing, we apply GenAI to find grammar errors and fix them.

References

[1] Vaibhav Adlakha, Shehzaad Dhuliawala, Kaheer Suleman, Harm de Vries, and Siva Reddy. 2022. TopiOCQA: Open-domain Conversational Question Answering with Topic Switching. *Transactions of the Association for Computational Linguistics* 10 (2022), 468–483.

[2] Krisztian Balog. 2021. Conversational AI from an Information Retrieval Perspective: Remaining Challenges and a Case for User Simulation. In *DESIRES (CEUR Workshop Proceedings, Vol. 2950)*. CEUR-WS.org, 80–90.

[3] Guanliang Chen, Jie Yang, Claudia Hauff, and Geert-Jan Houben. 2018. LearningQ: a large-scale dataset for educational question generation. In *Proceedings of the international AAAI conference on web and social media*. AAAI Press, 481–490.

[4] Haonan Chen, Zhicheng Dou, Kelong Mao, Jiongnan Liu, and Ziliang Zhao. 2024. Generalizing Conversational Dense Retrieval via LLM-Cognition Data Augmentation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2700–2718.

[5] Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2020. TREC CAsT 2019: The Conversational Assistance Track Overview. *CoRR* abs/2003.13624 (2020).

[6] Yubin Ge, Ziang Xiao, Jana Diesner, Heng Ji, Karrie Karahalios, and Hari Sundaram. 2023. What should I Ask: A Knowledge-driven Approach for Follow-up Questions Generation in Conversational Surveys. In *Proceedings of the 37th Pacific Asia Conference on Language, Information and Computation*. Association for Computational Linguistics, 113–124.

[7] Marcel Gohsen, Johannes Kiesel, Mariam Korashi, Jan Ehlers, and Benno Stein. 2023. Guiding Oral Conversations: How to Nudge Users Towards Asking Questions?. In *Proceedings of the 2023 Conference on Human Information Interaction and Retrieval*. ACM, 34–42.

[8] Shrey Gupta, Anmol Agarwal, Manas Gaur, Kaushik Roy, Vignesh Narayanan, Ponnurangam Kumaraguru, and Amit P. Sheth. 2022. Learning to Automate Follow-up Question Generation using Process Knowledge for Depression Triage on Reddit Posts. In *Proceedings of the Eighth Workshop on Computational Linguistics and Clinical Psychology*. 137–147.

[9] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685 [cs.CL] https://arxiv.org/abs/2106.09685

[10] Jiaxiong Hu, Jingya Guo, Ningjing Tang, Xiaojuan Ma, Yuan Yao, Changyuan Yang, and Yingqing Xu. 2024. Designing the Conversational Agent: Asking Follow-up Questions for Information Elicitation. *Proceedings of the ACM on Human-Computer Interaction* 8, CSCW1 (2024), 1–30.

[11] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. *CoRR* abs/2310.06825 (2023).

[12] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. 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*. Association for Computational Linguistics, 6769–6781.

[13] Kimmo Kettunen. 2014. Can type-token ratio be used to show morphological complexity of languages? *Journal of Quantitative Linguistics* 21, 3 (2014), 223–245.

[14] Johannes Kiesel, Volker Bernhard, Marcel Gohsen, Josef Roth, and Benno Stein. 2022. What is That? Crowdsourcing Questions to a Virtual Exhibition. In *Proceedings of the 2022 Conference on Human Information Interaction and Retrieval*. ACM, 358–362.

[15] Johannes Kiesel, Marcel Gohsen, Nailia Mirzakhmedova, Matthias Hagen, and Benno Stein. 2024. Simulating Follow-Up Questions in Conversational Search. In *European Conference on Information Retrieval (Lecture Notes in Computer Science, Vol. 14609)*. Springer, 382–398.

[16] Gangwoo Kim, Sungdong Kim, Byeongguk Jeon, Joonsuk Park, and Jaewoo Kang. 2023. Tree of clarifications: Answering ambiguous questions with retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 996–1009.

[17] Wei-Jen Ko, Te-Yuan Chen, Yiyan Huang, Greg Durrett, and Junyi Jessy Li. 2020. Inquisitive Question Generation for High Level Text Comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 6544–6555.

[18] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*. 611–626.

[19] 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*. 11526–11544.

[20] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 7871–7880.

[21] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.

[22] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437* (2024).

[23] Jianyu Liu, Yi Huang, Sheng Bi, Junlan Feng, and Guilin Qi. 2025. From Superficial to Deep: Integrating External Knowledge for Follow-up Question Generation Using Knowledge Graph and LLM. In *Proceedings of the 31st International Conference on Computational Linguistics*. Association for Computational Linguistics, 828–840.

[24] Siyang Liu, Sahand Sabour, Yinhe Zheng, Pei Ke, Xiaoyan Zhu, and Minlie Huang. 2022. Rethinking and refining the distinct metric. *arXiv preprint arXiv:2202.13587* (2022).

[25] Yuxuan Liu, Haipeng Liu, and Ting Long. 2024. HierLLM: Hierarchical Large Language Model for Question Recommendation. *CoRR* abs/2409.06177 (2024).

[26] Zhe Liu, Taekyu Kang, Haoyu Wang, Seyed Hossein Alavi, and Vered Shwartz. 2025. Bridging Information Gaps with Comprehensive Answers: Improving the Diversity and Informativeness of Follow-Up Questions. *arXiv preprint arXiv:2502.17715* (2025).

[27] Yan Meng, Liangming Pan, Yixin Cao, and Min-Yen Kan. 2023. FollowupQG: Towards information-seeking follow-up question generation. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 252–271.

[28] Fengran Mo, Yifan Gao, Chuan Meng, Xin Liu, Zhuofeng Wu, Kelong Mao, Zhengyang Wang, Pei Chen, Zheng Li, Xian Li, et al. 2025. UniConv: Unifying Retrieval and Response Generation for Large Language Models in Conversations. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 6936–6949.

[29] Fengran Mo, Kelong Mao, Ziliang Zhao, Hongjin Qian, Haonan Chen, Yiruo Cheng, Xiaoxi Li, Yutao Zhu, Zhicheng Dou, and Jian-Yun Nie. 2024. A survey of conversational search. *ACM Transactions on Information Systems* (2024).

[30] Fengran Mo, Kelong Mao, Yutao Zhu, Yihong Wu, Kaiyu Huang, and Jian-Yun Nie. 2023. ConvGQR: Generative Query Reformulation for Conversational Search. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

[31] 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)*. 4998–5012.

[32] Fengran Mo, Jian-Yun Nie, Kaiyu Huang, Kelong Mao, Yutao Zhu, Peng Li, and Yang Liu. 2023. Learning to relate to previous turns in conversational search. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1722–1732.

[33] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. In *CoCo@NIPS (CEUR Workshop Proceedings, Vol. 1773)*. CEUR-WS.org.

[34] OpenAI. 2023. GPT-4 Technical Report. *CoRR* abs/2303.08774 (2023).

[35] Boyuan Pan, Hao Li, Ziyu Yao, Deng Cai, and Huan Sun. 2019. Reinforced Dynamic Reasoning for Conversational Question Generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2114–2124.

[36] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 311–318.

[37] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems* 36 (2023), 53728–53741.

[38] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of machine learning research* 21 (2020), 140:1–140:67.

[39] Md. Rashad Al Hasan Rony, Soumya Ranjan Sahoo, Abbas Goher Khan, Ken E. Friedl, Viju Sudhi, and Christian Süß. 2024. Incorporating Query Recommendation for Improving In-Car Conversational Search. In *European Conference on Information Retrieval (Lecture Notes in Computer Science, Vol. 14612)*. Springer, 304–312.

[40] Pooja Rao SB, Manish Agnihotri, and Dinesh Babu Jayagopi. 2020. Automatic follow-up question generation for asynchronous interviews. In *Proceedings of the Workshop on Intelligent Information Processing and Natural Language Generation*. 10–20.

[41] Ming-Hsiang Su, Chung-Hsien Wu, and Yi Chang. 2019. Follow-Up Question Generation Using Neural Tensor Network-Based Domain Ontology Population

in an Interview Coaching System.. In *INTERSPEECH*. 4185–4189.

[42] Ming-Hsiang Su, Chung-Hsien Wu, Kun-Yi Huang, Qian-Bei Hong, and Huai-Hung Huang. 2018. Follow-up Question Generation Using Pattern-based Seq2seq with a Small Corpus for Interview Coaching.. In *INTERSPEECH*. 1006–1010.

[43] Yuchong Sun, Che Liu, Kun Zhou, Jinwen Huang, Ruihua Song, Xin Zhao, Fuzheng Zhang, Di Zhang, and Kun Gai. 2024. Parrot: Enhancing Multi-Turn Instruction Following for Large Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 9729–9750.

[44] Guan Wang, Sijie Cheng, Xianyuan Zhan, Xiangang Li, Sen Song, and Yang Liu. 2024. OpenChat: Advancing Open-source Language Models with Mixed-Quality Data. In *The Twelfth International Conference on Learning Representations*. OpenReview.net.

[45] Jian Wang and Wenjie Li. 2021. Template-guided clarifying question generation for web search clarification. In *Proceedings of the 30th ACM international conference on information & knowledge management*. 3468–3472.

[46] Tianlu Wang, Ping Yu, Xiaoqing Ellen Tan, Sean O'Brien, Ramakanth Pasunuru, Jane Dwivedi-Yu, Olga Golovneva, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. 2023. Shepherd: A critic for language model generation. *arXiv preprint arXiv:2308.04592* (2023).

[47] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in neural information processing systems*, Vol. 35. 24824–24837.

[48] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024. Qwen2.5 Technical Report. *CoRR* abs/2412.15115 (2024).

[49] Liu Yang, Hamed Zamani, Yongfeng Zhang, Jiafeng Guo, and W. Bruce Croft. 2017. Neural Matching Models for Question Retrieval and Next Question Prediction in Conversation. *CoRR* abs/1707.05409 (2017).

[50] 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*. 418–428.

[51] Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. 2024. ChatGLM: A Family of Large Language Models from GLM-130B to GLM-4 All Tools. *CoRR* abs/2406.12793 (2024).

[52] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675* (2019).

[53] Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024. WildChat: 1M ChatGPT Interaction Logs in the Wild. In *The Twelfth International Conference on Learning Representations*. OpenReview.net.

[54] Ziliang Zhao, Zhicheng Dou, Yu Guo, Zhao Cao, and Xiaohua Cheng. 2023. Improving search clarification with structured information extracted from search results. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3549–3558.

[55] Ziliang Zhao, Zhicheng Dou, Jiaxin Mao, and Ji-Rong Wen. 2022. Generating clarifying questions with web search results. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 234–244.

[56] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric P. Xing, Joseph E. Gonzalez, Ion Stoica, and Hao Zhang. 2024. LMSYS-Chat-1M: A Large-Scale Real-World LLM Conversation Dataset. In *The Twelfth International Conference on Learning Representations*. OpenReview.net.

[57] Yaowei Zheng, Richong Zhang, Junhao Zhang, YeYanhan YeYanhan, and Zheyan Luo. 2024. LlamaFactory: Unified Efficient Fine-Tuning of 100+ Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*. 400–410.