



# ClariLM: Enhancing Open-domain Clarification Ability for Large Language Models

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

Jian Xie  
Baichuan AI  
Beijing, China  
xiejian1990@gmail.com

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

## Abstract

Active understanding and clarification of user intent is crucial for information-seeking systems based on Large Language Models (LLMs), as it enhances search efficiency and improves user experience for human-LLM interaction. While existing systems rely on domain-specific resources to generate clarifying questions, they face challenges when extended to open-domain scenarios due to the lack of human-LLM clarification data. In this paper, we propose *ClariLM* to synthesize large-scale clarification data and enhance the LLMs' clarification capability. Specifically, we design two key stages to prepare data: first, given a user question, the Clarification Facet Detection (CFD) stage employs a facet mining model learned from human-LLM conversation logs to predict realistic potential clarification candidates. Additionally, it incorporates direct predictions from powerful LLMs as supplements to guarantee comprehensive facet coverage. While CFD ensures high recall of facet candidates, the subsequent Optimal Facet Selection (OFS) stage synthesizes a set of new questions and employs a reasoning model to annotate the optimal facet for each question, which further improves the precision of *ClariLM* in clarification necessity prediction and optimal facet selection. The collected data are then applied for supervised fine-tuning, followed by constructing preference data for preference optimization. Experiments on our custom test set and two public benchmarks demonstrate that *ClariLM* significantly outperforms various baseline models across clarification necessity, clarifying question quality, and GPT-4-based comparative evaluation.

## CCS Concepts

• Information systems → Language models.

## Keywords

Clarifying Question, Large Language Model, Conversational Information Seeking, Conversation Log

\*Zhicheng Dou is the corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

CIKM '25, Seoul, Republic of Korea

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.  
ACM ISBN 979-8-4007-2040-6/2025/11  
<https://doi.org/10.1145/3746252.3761068>

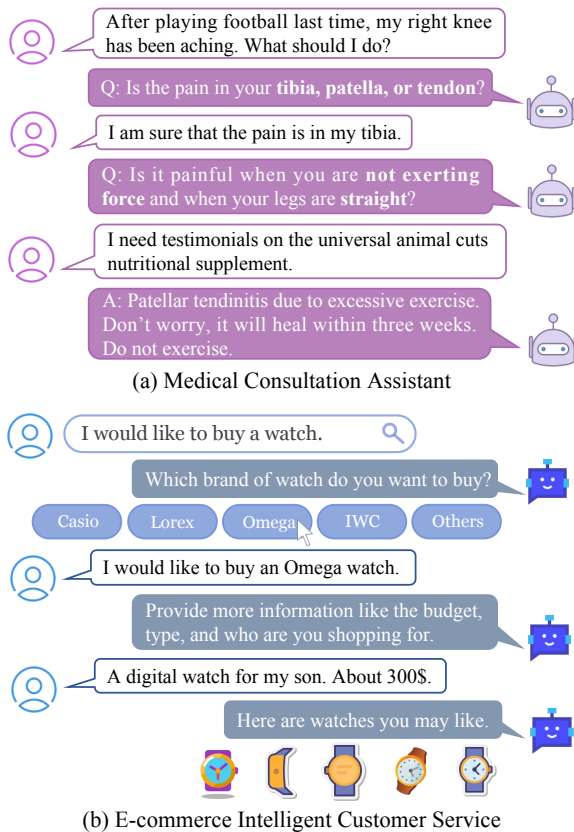
## ACM Reference Format:

Ziliang Zhao, Haonan Chen, Shiren Song, Jian Xie, and Zhicheng Dou. 2025. ClariLM: Enhancing Open-domain Clarification Ability for Large Language Models. 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/10.1145/3746252.3761068>

## 1 Introduction

In conversational information seeking [27–30] scenarios where users interact with Large Language Models (LLMs), the user questions are often incomplete or ambiguous [4], with multi-dimensional missing information in a single utterance [55]. In such cases, an LLM-initiated clarifying question [48] can help explore the user's real intent and improve the user's interaction experience. Clarification is useful in many scenarios. For example, in Figure 1, case (a) illustrates a medical consultation assistant [24]. The user states that their right knee has been hurting after playing football. The system sequentially guides the user to provide more details by asking about the specific pain part and whether it hurts when extending the leg. This helps further identify the possible symptoms and propose appropriate solutions. Besides, case (b) shows an e-commerce customer service assistant [14]. The user expresses interest in buying a watch, and the system first provides a list of brands, then inquires about budget preferences, watch type, and whether the purchase is for someone else. This enables more precise recommendations. These clarifying questions help progressively refine user intent, enhancing user experience while efficiently fulfilling their needs.

Recently, many studies on clarification design how to generate high-quality clarifying questions within specific scenarios, such as Web search [49, 50, 56, 57], conversational search [1–3], and knowledge-intensive QA/dialogue [16, 51]. However, these methods are typically tied to specific tasks and domains, while **it is challenging to guide the clarification ability of LLMs in a generalized human-LLM interaction scenario due to the lack of data**. Specifically, **first**, existing studies usually rely on domain-specific resources to obtain clarification data. For example, Web search clarification data can be obtained through query reformulation, while in conversational search, the quality of clarification can be judged by improvements in search results. However, in a human-LLM interaction scenario, it is difficult to obtain intent clarification signals [4], leading to **the lack of original intent clarification data**. **Second**, a single query often has multiple facets requiring clarification [55], therefore several clarifying questions



**Figure 1: Two examples of asking clarifying questions in conversational information-seeking systems.**

are reasonable as candidates. For example, when a user asks an LLM to recommend nearby coffee shops, it is necessary to first clarify the user’s location before addressing other details, such as the type of coffee or the number of people going. Therefore, it is also challenging to determine which facet should be targeted for questioning or whether a clarifying question should be asked due to **the lack of clarification preference data**.

In this paper, we propose **ClariLM** to enhance the clarification ability of LLMs. To mitigate the data lacking issue for ClariLM, we design a two-stage detection-selection framework that captures realistic and comprehensive user intents and decides whether to ask a clarifying question and which missing intent to target. In the first stage **Clarification Facet Detection (CFD)**, the goal is to obtain a both *realistic* and *comprehensive* set of clarification facet candidates, improving recall in clarification targeting. To ensure the realism of clarification, we propose incorporating human-LLM conversation logs [33], mining user rephrasing or specifications of their intent within a session, which could be active clarification behaviors. This information is used to train a log facet miner, which is then applied to predict potential clarification facets. On the other hand, using only the log facet miner suffers from sparsity issues, which means that the predicted facets can be biased toward conversation log data. To address this, we prompt and guide powerful LLMs (such as GPT-4o) to directly generate multiple facet candidates as a supplement. Finally, we combine the results of the two above facet miners to

compose a set of potential clarification facets and corresponding clarifying questions candidates for a given user question.

In the second stage **Optimal Facet Selection (OFS)**, we aim to build clarification targets by selecting the best clarification facet from the CFD-generated candidates to ask the user or decide to directly answer the user’s question without clarification. Building upon CFD, this step improves the clarification precision. Due to the lack of annotated data, choosing which facet to ask about or deciding whether to ask a question is challenging. To this end, we employ a large reasoning model to annotate the most important clarification facet or choose to answer directly. The annotated dataset is then applied for supervised fine-tuning and evaluation of ClariLM. To further enable ClariLM to explicitly learn how to determine whether to ask questions and based on which facets to ask, we define three preference types: (1) Asking a question is better than responding directly, (2) Responding directly is better than asking a question, (3) One question is better than another question. We then collect preference data based on the annotated dataset and three preference types and train ClariLM with Direct Preference Optimization [34] to further improve its clarification ability.

In our experiments, we first evaluate ClariLM on two open-domain LLM clarification benchmarks: CLAMBER [52], a recently proposed general LLM clarification benchmark for evaluating open-domain clarification ability for LLMs. IN3 [33], which focuses on clarification in agent-based interactions. Additionally, we use a portion of our constructed dataset as a held-out test set. On all three test sets, we evaluate both clarification necessity and clarifying question quality together with GPT-4-based comparative evaluation and compare ClariLM with multiple groups of state-of-the-art LLM baseline models. Experimental results show that ClariLM outperforms existing LLMs in user intent mining and question-asking ability, effectively requesting clarification when user questions contain missing information instead of providing direct responses. Moreover, ClariLM exhibits better decision-making in whether to ask a question when clarification is unnecessary.

To sum up, our contributions include:

- To our best knowledge, ClariLM is the first method of studying how to clarify user intent in human-LLM conversations by leveraging LLM logs.
- To build large-scale LLM clarification data, we propose a novel two-stage approach to generate multiple facet candidates and select the optimal clarification facet.
- The experimental results demonstrate the effectiveness of ClariLM across multiple benchmarks.

## 2 Related Work

### 2.1 Clarification for Information Seeking

Search clarification has become one of the recent research hotspots. This line of research primarily focuses on two scenarios: conversational dense retrieval and Web search. In conversational dense retrieval, the system’s goal is to retrieve more accurate documents for users by asking clarifying questions. For example, Aliannejadi et al. [4] first design a framework for conversational dense retrieval clarification consisting of question retrieval, question selection, and document retrieval. Besides, Hashemi et al. [11] propose Guided Transformer to help select the optimal clarifying question

by leveraging external retrieved documents. Bi et al. [6] further apply negative feedback to generate yes/no-formed questions to explore the user’s search intent within the several last conversation turns. However, selecting clarifying questions from a fixed-size question pool cannot satisfy the users’ complex search intents in real-world scenarios. Thus, a much more preferable paradigm is to generate a clarifying question based on the complex real-world information needs Aliannejadi et al. [1], Krasakis et al. [17], Mass et al. [26], Sekulic et al. [38], Wang et al. [42, 43]. As for the Web search, the system goal is to generate a clarifying question together with several candidate facets for ambiguous or faceted Web short queries [48]. Some studies focus on how to generate informative and effective questions [57] and another series of studies focus on generating high-quality facets [12, 13, 37, 56].

## 2.2 Clarification for Large Language Models

Recently, with the development of Large Language Models (LLMs), there are also some studies on how to let LLMs ask clarifying questions in the interaction with online users. For example, in clarification for conversational retrieval, LLMs can be used to generate clarifying questions and simulate user responses [42, 43]. For the web search clarification, the LLMs can be applied to generate better clarifying questions or query facets [19, 23, 32, 55]. In conversational recommender systems, leveraging LLMs can also improve the recommendation effectiveness [10, 41]. These studies have shown that LLMs help improve the system’s performance due to LLMs’ strong natural language generation and instruction-following capability in many real-world application scenarios. With the powerful natural language generation capabilities of LLMs, the existing tasks and scenarios can be effectively improved [31]. However, it is still unknown whether LLMs can be used to expand the existing clarification scenarios so that clarification can have an open-domain application range. To this end, many LLM-based datasets and methods have been proposed. For example, Zhang et al. [52] built an open-domain LLM clarification benchmark CLAMBER for evaluating the LLMs’ clarification ability. Li et al. [22] elicit user interest by asking clarifying questions generated by LLMs. Andukuri et al. [5] further design a reinforcement learning system to generate better clarifying questions. Deng et al. [9] propose a “Rephrase and Respond (RaR)” framework to guide the LLM clarification.

## 2.3 Other Clarification Scenarios

Beyond conversational and Web search, clarification mechanisms have been extensively studied across various domains to enhance human-computer interaction. In Question Answering (QA) systems, a significant series of research [18, 35, 36, 39] has focused on developing techniques for generating clarifying questions. These approaches aim to resolve ambiguities [40] in user queries by engaging in targeted dialogue, thereby improving the precision and relevance of system responses. Besides, the field of Conversational Recommender Systems (CRS) [7, 8, 10, 45, 53, 59, 60] has similarly adopted clarification strategies to refine user preferences and deliver more personalized recommendations. These systems often employ multi-turn dialogues to elicit additional user feedback or clarify ambiguous requests, particularly in complex recommendation scenarios where user intent may not be immediately apparent [20, 21].

Recent advances have focused on integrating reinforcement learning and user modeling techniques to make the clarification process more natural and efficient [41]. Emerging research in multi-modal clarification [44, 46, 47] extends these concepts to scenarios involving visual, auditory, or other sensory inputs. These systems face unique challenges in interpreting and disambiguating multi-modal user queries, where clarification may be needed to resolve conflicts between different input modalities or to supplement missing information. For example, when processing a query that combines both text and images, the system might need to ask follow-up questions to determine which aspects of the visual input are most relevant to the user’s information need. This line of work represents an important frontier in making AI systems more robust and adaptable to real-world, multi-modal interactions.

## 3 ClariLM

The construction of open-domain LLM clarification data consists of two main stages: (1) Clarification Facet Detection (CFD), which generates a set of candidate facets (and corresponding clarifying questions) for a given user query based on human-LLM conversation logs and strong LLMs, and (2) Optimal Facet Selection (OFS), which selects the best facet or answer directly to form a large-scale dataset. The dataset is then applied for the training and evaluation of ClariLM. In this section, we first formalize the clarification task, then detail the CFD and OFS components, including their algorithmic design and training methodology.

### 3.1 Problem Formulation

We consider an open-domain clarification task for an LLM-based assistant. The interaction begins with a user question  $q$  (which may be a question, instruction, or search query) that could be ambiguous or under-specified. The assistant must produce an output  $r$  that either clarifies the query by asking a clarifying question  $c$  to elicit the missing information (if the query is found to be ambiguous) or answers the query directly with a factual or task-completing response  $a$  (if the query is sufficiently clear or after receiving clarification). Thus, the assistant faces a decision problem: whether to ask for clarification or not, and if yes, what specifically to ask:

$$\text{ClariLM}(q) = r \in \{c, a\} \text{ where } \begin{cases} c & \text{for clarifying.} \\ a & \text{for answering.} \end{cases} \quad (1)$$

### 3.2 Clarification Facets Detection (CFD)

The objective of CFD is to generate as many realistic and comprehensive clarification facet candidates as possible for user input questions to improve recall. The full workflow of CFD is illustrated in the left part of Figure 2. As shown, CFD consists of two models: FM-LOG and FM-LLM. FM-LOG extracts user behaviors related to intent clarification within a conversational session from conversation logs. These behavioral data are used to perform supervised fine-tuning on FM-LOG. Ultimately, this model can generate multiple potential clarification facets for a given user question through sampling. By learning from real-world conversation logs, FM-LOG captures authentic user clarification facets and their corresponding question candidates. However, due to the sparsity of conversation logs, FM-LOG may occasionally produce biased results, particularly



**Figure 2: The ClariLM framework consists of Clarification Facets Detection (CFD) and Optimal Facet Selection (OFS).**

for long-tail questions. To address this limitation, CFD additionally incorporates FM-LLM, which leverages carefully designed prompts to guide powerful Large Language Models (e.g., GPT-4) in generating as many clarification facets and corresponding questions as possible. Finally, the outputs from FM-LOG and FM-LLM are merged and deduplicated to produce the final results of CFD.

**3.2.1 FM-LOG Clarification Data Obtaining.** The objective of FM-LOG is to leverage user-initiated clarification data from conversation logs to generate authentic, user-interest-aligned clarification facet candidates for new user questions. To achieve this, it is first necessary to collect a large volume of real human-LLM conversation logs. In this paper, we merge two mainstream released log collections, ShareGPT<sup>1</sup> and WildChat [54], as our logs  $\mathcal{L}$ . However, extracting user-initiated clarification data from these logs is challenging. To address this, we employ a powerful LLM to extract useful information from  $\mathcal{L}$ . We design a prompt structure, as illustrated in Figure 3, to extract data in the format of  $(q_i, c, q_{i+1}, f)$ , where  $q_i$  and  $q_{i+1}$  represent the user’s  $i$ -th and  $(i+1)$ -th consecutive questions to the system, while  $c$  and  $f$  denote the LLM-generated clarification question and facet for the transition from  $q_i$  to  $q_{i+1}$ .

Specifically, first, we declare our task as extracting data where the user’s consecutive questions  $q_i$  and  $q_{i+1}$  involve the latter clarifying the former, and we instruct the LLM to generate useful results such as the clarification question  $c$ . We then provide three positive demonstrations where  $q_{i+1}$  represents an active user clarification of  $q_i$ . On the other hand, considering that many instances of  $q_{i+1}$  do not clarify  $q_i$  (e.g., due to intent shifts, follow-up questions, etc.), we further include four negative demonstrations to enhance the model’s annotation accuracy. Finally, we input the entire sequence of user questions from a log session and instruct the model to output the results in a specific JSON format.

**3.2.2 FM-LOG Training.** After data collection, we obtained 104k instances of data in the format  $D = (q_i, c, q_{i+1}, f)$ . We further process the data such that the input is  $q_i$ , and the output is the combination of  $f$  and  $c$ . We select LLaMA3-8B<sup>2</sup> as our base model and perform Supervised Fine-Tuning (SFT) on the collected data  $D$  to obtain the FM-LOG model with strong clarification prediction ability. Formally, FM-LOG aims to optimize the following loss function:

$$\mathcal{L}_{\text{FM-LOG}}(\theta) = - \sum_{(q_i, c, f) \in \mathcal{D}} \log p(f, c | q_i; \theta), \quad (2)$$

where  $\theta$  denotes the model parameters, and  $P_\theta(\cdot | q_i)$  represents the conditional probability of generating the clarification question  $c$  and facet  $f$  given the input query  $q_i$ . This objective ensures that FM-LOG learns to generate both clarifying questions and facets that align with user interests based on historical interaction patterns extracted from human-LLM conversation logs.

**3.2.3 FM-LOG Inference.** Since a single user question  $q_i$  may correspond to multiple clarification facets  $f$  (or clarifying questions  $c$ ), while the training phase only generates one facet per question, we employ a sampling technique during inference to bridge this gap. Specifically, we generate  $k$  facet candidates for each user question through beam search together with top-k and top-p sampling. Subsequently, we remove all facets that are completely identical to others to ensure diversity in the sampled results.

Formally, during inference, given an input question  $q_i$ , FM-LOG produces a set of candidate facets  $\{f_1, f_2, \dots, f_k\}$  and a set of corresponding clarifying questions  $\{c_1, c_2, \dots, c_k\}$  by sampling from the model’s output distribution  $P_\theta(f, c | q_i)$ . We then apply deduplication to eliminate redundant facets, retaining only distinct candidates for downstream processing. This approach enhances coverage of potential user interests while maintaining high-quality, diverse clarifications. Mathematically, we denote the set of facets obtained

<sup>1</sup>[https://huggingface.co/datasets/anon8231489123/ShareGPT\\_Vicuna\\_unfiltered](https://huggingface.co/datasets/anon8231489123/ShareGPT_Vicuna_unfiltered)

<sup>2</sup>LLaMA3-8B: <https://huggingface.co/meta-llama/Meta-Llama-3-8B>

· The user often asks consecutive questions during interactions with the LLM. After the user asks  $q^i$ , the LLM sometimes ask a clarifying question  $c$ , and the user rearticulate her/his intent by reformulating  $q^i$  as  $q^{i+1}$ .

· In this process,  $q^{i+1}$  has **the same target or intent** with  $q^i$ , but  $q^{i+1}$  adds some constraints or conditions or rewrite  $q^i$  as clarification.

· Here are some examples:  
**Example 1:**  
 $q^i$ : "I wanna buy a watch for my sister. Could you recommend for me?",  
 $c$ : "Sure, but could you please tell me whom are you shopping for?",  
 $q^{i+1}$ : "I would like to buy a Casio or Rolex watch.",  
 $f$ : "specifying the brand to purchase".  
*{More positive examples}*

And here are some bad examples where  $q^{i+1}$  is NOT the clarification of  $q^i$ :  
**Negative Example 1** (shifts the original intent):  
 $q^i$ : "Explain the different types of registers available in ARM",  
 $q^{i+1}$ : "what is the msr instruction in ARM used for"  
*{More negative examples}*

· Now, given a sequence of questions asked by the user to the LLM: *{Questions}*  
 · Please refer to the examples above and find potential clarification processes between two consecutive questions  $q^i$  and  $q^{i+1}$ , then write proper clarifying question  $c$  and its corresponding clarification facet  $f$ , and output according to the following JSON format:  
*{JSON format}*

Note:  
 - The  $q^i$  and  $q^{i+1}$  can be rewritten to include complete intent respectively.  
 - If multiple clarification processes are found, generate multiple instances. If no clarification is found, simply output an empty list.  
 - The identification of clarification is very strict. Any instance where  $q^{i+1}$  deviates from the intent of  $q^i$  cannot be considered a clarification and should therefore be discarded. Only when  $q^{i+1}$  introduces new conditions based on  $q^i$  can it be regarded as a clarification. Therefore, be as cautious as possible when generating the output. Just output the result based on the JSON format and do not output anything else. Try your best!

**Figure 3: Prompt of FM-LOG extracting clarification data.**

from the external/log-based source as:

$$F_{\log}(x) = \text{Sampling}(\text{FM-LOG}(x)) = \{f_1^{\log}, f_2^{\log}, \dots, f_m^{\log}\}. \quad (3)$$

**3.2.4 FM-LLM.** FM-LOG utilizes only conversation logs as the sole training data source. While this approach effectively captures users' genuine intents, it may occasionally suffer from overfitting issues, resulting in the generation of insufficiently comprehensive facets. However, in reality, user intents are highly diverse. To achieve the objective of CFD—maximizing the recall of generated clarifying facets—it is essential not only to ensure realism but also to generate facets that are as comprehensive as possible. To this end, we propose FM-LLM as a complementary method to FM-LOG, as illustrated in Figure 2 (b). We design prompts to leverage the generative capabilities of powerful LLMs to brainstorm potential clarification facet candidates along with their corresponding clarifying questions. This LLM-generated facet list can introduce interpretations that may be overlooked in the logs, particularly for less common user queries or creative ambiguities, by harnessing the model's world knowledge and reasoning capabilities. Formally,

$$F_{\text{llm}}(x) = \{f_1^{\text{llm}}, f_2^{\text{llm}}, \dots, f_n^{\text{llm}}\}. \quad (4)$$

Each  $f$  represents a candidate facet (also clarifying question<sup>3</sup>) in natural language form (often phrased as a short noun or clause, e.g. "which device", "the country or the person").

<sup>3</sup>Each facet corresponds to a clarifying question and vice versa.

**3.2.5 Result Fusion.** After obtaining  $F_{\log}(x)$  and  $F_{\text{llm}}(x)$ , we then perform a fusion of candidate facets from the two sources to form the final  $F(x)$ . The fusion process involves taking the union of the sets and resolving duplicates or semantically overlapping facets:  $F(x) = \text{dedup}(F_{\log}(x) \cup F_{\text{llm}}(x))$ . We denote the final distinct facets as:  $F(x) = \{f_1, f_2, \dots, f_k\}$ .

After fusion, we obtain a set  $F(x)$  of candidate facets that represent the different clarification questions the LLM could ask. At this stage, **we are agnostic as to which facet (if any) is actually necessary**. We aim for high recall, that is,  $F(x)$  should contain the truly relevant missing piece if there is one. The next stage (Optimal Facet Selection, OFS) will determine which facet to actually use. Importantly, the CFD stage can be seen as generating multiple possible clarifying questions: each facet  $f_i$  corresponds to a question template like "Could you clarify [ $f_i$ ]?" or "What did you mean regarding [ $f_i$ ]?". In our implementation, we allow the language model to phrase the question naturally, but  $f_i$  guides the content of that question. By separating content (facet) from phrasing, we ensure that the model focuses on the right topic when asking a question.

### 3.3 Optimal Facet Selection (OFS)

After collecting a candidate set of facets for a question in the CFD stage, selecting the most appropriate facet to present to the user poses a significant challenge. First, it is important to note that different facets (or clarification questions) provide varying levels of informational value. For example, given the query "Give me a list of good coffee shops?", Q1: "Could you specify the location or city where you are looking for good coffee shops?" and Q2: "Are you visiting alone or with a group? If with a group, how many people will be joining you?". Clearly, the first question addresses a crucial aspect (providing substantial informational gain), while the second is largely irrelevant (offering minimal informational value). Additionally, some user questions are sufficiently clear and do not require further clarification, warranting a direct response. Therefore, determining whether to clarify or directly answer is essential in real-world applications.

To this end, when an LLM receives a user question, it should learn when to clarify, when to answer directly, and if clarifying, which facet to prioritize. Thus, the OFS stage aims to determine the optimal strategy for each user question: either posing a clarification question for a specific facet or choosing to respond directly without clarification. OFS addresses this by training ClariLM on constructed data. While CFD focuses on improving recall, OFS further enhances precision. Specifically, OFS first collects a set of user questions, including the first-question set  $Q^L$  as question seed from conversation logs in each session, as well as a newly generated question set  $Q^G$  obtained by analyzing various dimensions of user questions and prompting a powerful LLM for generation. Similar to the CFD stage, combining  $Q^L$  and  $Q^G$  ensures both realism and comprehensiveness in the question set. After obtaining the question set  $Q = (Q^L, Q^G)$ , we execute the CFD process mentioned above to generate clarification facet and corresponding question candidates  $F(q)$  for each  $q \in Q$  using the combination of FM-LOG and FM-LLM. Subsequently, due to the lack of labeled data on whether to clarify and which clarification is optimal, we employ a reasoning model to analyze all clarification candidates (including direct answers) with

chain-of-thoughts and select the best one. Finally, the annotated data is used for supervised fine-tuning (SFT) and direct preference optimization (DPO) to train the final ClariLM model.

**3.3.1 Obtaining  $Q^L$ .** By incorporating the initial user question in each session from human-LLM conversation logs, the model can more effectively learn real-world user query patterns and corresponding clarification strategies, as illustrated in Figure 2 (c1). However, due to noise and the nature of conversation log construction, many initial questions are actually context-dependent but appear truncated. To address this, we designed prompts for a powerful LLM to filter and retain only high-quality context-independent questions in  $Q^L$  while discarding other extremely long, non-English, and single-turn questions. After this processing, we obtained approximately 11k high-quality **seed questions** as  $Q^L$ .

**3.3.2 Obtaining  $Q^G$ .** The data amount of  $Q^L$  is still insufficient for ClariLM training. To address the potential sparsity issue in  $Q^L$  and enhance question diversity, we further propose a systematic approach for synthesizing another comprehensive question set  $Q^G$  through multi-dimensional optimization as the compensation for  $Q^L$ . As illustrated in Figure 2 (c2), our method expands question diversity across three critical dimensions. First, **topic diversity** is achieved by employing distinct persona profiles that guide language models to generate questions spanning various domains and perspectives. This persona-driven approach ensures coverage of topics ranging from technical subjects to everyday scenarios. Second, we implement **enhanced in-context learning** by strategic sampling from multiple established datasets including  $Q^L$ , CLAMBER, IN3, and prominent QA resources such as AmbigQA, Natural Questions (NQ), and PopQA. This cross-dataset prompting mechanism enables the model to assimilate diverse questioning patterns and knowledge structures. Third, we introduce **controlled clarification tuning**, where generated questions undergo intentional information adjustment: 80% retain their original form, 10% are enriched with additional contextual details, and 10% are simplified through information reduction. This calibrated approach balances question complexity while expanding the model’s capacity to handle varying information densities. The process is directly accomplished by a powerful LLM. We finally apply deduplication, resulting in a refined dataset of approximately 124k distinct synthetic questions. This multi-stage generation framework not only compensates for  $Q^L$ ’s limitations in coverage but also creates novel question formulations that bridge the gap between existing resources.

**3.3.3 Data Annotation.** Our data annotation framework initiates by leveraging the previously discussed FM-LOG and FM-LLM mechanisms from the CFD phase to generate clarification candidates for user questions from  $Q = (Q^L \cup Q^G)$ . Formally, for each user question  $q$ , we construct a candidate set  $C = \{c_1, c_2, \dots, c_n\} \cup a$ , where  $c_i$  represent clarification questions and  $a$  denotes the original answer incorporated as an alternative response option. This formulation acknowledges that abstaining from clarification constitutes a valid system decision and improves user experiences.

Conventional information gain-based approaches typically require annotated  $(x, y)$  pairs to compute comparative metrics like  $E[\log p(y|x_1)] - E[\log p(y|x)]$  versus  $E[\log p(y|x_2)] - E[\log p(y|x)]$  for candidate selection. While effective in constrained domains

such as QA systems or conversational recommendation tasks with sufficient labeled data, these methods face critical limitations in general-purpose scenarios due to data sparsity and poor generalization capability. To address this challenge, we propose a novel **Chain-of-Thought Annotation Protocol (CoTAP)** employing Large Reasoning Models (LRMs) like DeepSeek-R1. The LRM processes instruction-templated inputs containing candidates  $C$  to generate a chain-of-thought  $t$ , followed by optimal candidate selection:  $\text{argmax}_{c \in C} P(c|t, x)$ , effectively distilling implicit knowledge through structured reasoning simulations.

Building upon this automated annotation framework, we further construct preference-optimized training data to enhance the model’s alignment with human communication patterns. Drawing on recent advances in direct preference optimization (DPO), we define three preference paradigms shown in Figure 2 (d):

- **Clarify  $\succ$  Answer:** For user questions that are ambiguous or under-specified, a clarifying question is better than a direct answer. For example, given the question “Tell me about Jordan”, asking “Do you mean the country or the person?” ( $q$ ) is better than directly giving an answer ( $a$ ) “Jordan is a country ...”, because “Jordan” in the question is ambiguous. This preference type teaches the model to not guess when it’s uncertain but to clarify instead.
- **Answer  $\succ$  Clarify:** For questions that are sufficiently specific, asking an unnecessary question would annoy the user or slow down the interaction. For instance, if the user asks “What is the capital of France?”, a direct answer “Paris” is preferred, and a clarifying question like “Do you mean the country France?” is not needed and is worse. This teaches the model to avoid asking questions when not needed.
- **Which Question (Facet) is Better ( $q_1 \succ q_2$ ):** For an ambiguous question with multiple possible facets, there is often an optimal facet to clarify first. For example, for the question “I need a visa”, possible facets might be “visa for which country?” instead of “what type of visa?”. If one of these facets is more crucial, the model should ask that first. These comparisons train the model to rank facets by usefulness. In practice, if multiple clarifications are necessary, the model could ask sequentially, but our immediate task is to pick the best single question to ask first.

In summary, from these processes, we assemble a large set of pairwise preferences. Each data point can be represented as  $D = (q, r^+, r^-)$  meaning for question  $q$ , output  $r^+$  is preferable to  $r^-$ . Here,  $r$  can be either a clarifying question or an answer. The statistics show that among the 140k pieces of data, 48,721 pieces of  $r^+$  are direct answers, 21,975 pieces are clarifying questions provided by FM-LOG, and 56,786 pieces are clarifying questions from FM-LLM. The statistical results further demonstrate that both FM-LOG and FM-LLM components in our proposed CFD module are capable of generating high-quality clarifying questions.

**3.3.4 Supervised Fine-Tuning (SFT).** After data annotation, we first perform SFT on ClariLM using the  $(q, r^+)$  pairs from dataset  $D$ , aiming to establish the model’s preliminary proficiency in both formatting clarifications and deciding whether to ask. Formally, the

**Table 1: Conversational Log Statistics**

	ShareGPT	WildChat
@Session	~79k	~838k
Session Ave. Length	3.07	4.67
Initial question Ave. Length	54.53	218.58

SFT phase optimizes the following loss function:

$$\mathcal{L}_{\text{SFT}}(\theta) = - \sum_{(q, r^+) \in \mathcal{D}} \log p(r^+ | T(q); \theta), \quad (5)$$

where  $T(\cdot)$  is the template. We also include a ‘‘Need Clarification’’ part in  $T(\cdot)$  to quickly judge whether ClariLM outputs a clarifying question or a direct answer.

**3.3.5 Direct Preference Optimization (DPO).** After SFT, the model can perform the task, but it might not perfectly obey preferences in edge cases or unseen questions, and might still be biased. For example, it might over-ask or under-ask depending on the data distribution. Therefore, we further fine-tune the model using the preference pairs and the DPO algorithm. DPO [34] is a technique to train LLMs from preference data in a direct, stable way, without needing a separate reward model as in traditional RLHF. The idea is to treat the preference comparison as a binary classification problem: given  $(x, y^+, y^-)$ , the model should assign a higher probability to  $y^+$  than  $y^-$ . In our task, DPO trains ClariLM to maximize a simple pairwise logistic objective:

$$\mathcal{L}_{\text{DPO}}(\theta) = - \sum_{(q, r^+, r^-) \in \mathcal{D}} \log \sigma \left( \beta \log \frac{p(r^+ | q; \theta)}{p(r^- | q; \theta)} \right), \quad (6)$$

where  $\sigma$  is the sigmoid function. In practice, we average this loss over all preference samples. During DPO training, we maintain the model’s ability to generate coherent questions/answers by initializing from the SFT model and by occasionally mixing in the supervised examples as additional training signals (a form of reward modeling regularization). We also include a mild KL-divergence regularization towards the SFT model’s distribution to prevent the model from drifting too far (a common practice in preference fine-tuning to retain base knowledge). After SFT-DPO training, we obtain the final ClariLM model. At inference time, ClariLM briefly takes a user question  $q$  and the instruction as the input. It then outputs the clarifying question or directly produces an answer.

## 4 Experiments

### 4.1 Conversation Logs

We use a combination of ShareGPT and WildChat [54] as the source of human-LLM conversation logs. ShareGPT is a website where users share their chat logs with ChatGPT, collecting a large number of real user-ChatGPT conversations. We obtain approximately 80k conversation sessions. WildChat is a recently released dataset containing over 1 million user-ChatGPT conversations, covering more than 2.5 million interactions. Compared to other conversation log datasets, WildChat offers a larger scale and supports multiple languages. Detailed statistics are shown in Table 1. It can be observed that in terms of session count and average session length, WildChat significantly outperforms ShareGPT.

### 4.2 Datasets

We randomly sample 2k pieces of data from our synthetic data  $Q$  as the test set denoted as *Our-test*<sup>4</sup>. Additionally, we evaluate our model on two publicly available datasets. We do not use any training data of the datasets, so the two datasets are actually out-of-domain test data. The first dataset is CLAMBER [52], a benchmark designed to assess LLMs’ ability to recognize and clarify ambiguous user information needs. Researchers constructed approximately 12,000 high-quality data points to evaluate the strengths, weaknesses, and potential risks of various off-the-shelf LLMs. The study found that current LLMs have limited practical utility in identifying and clarifying ambiguous user queries—even with techniques like Chain-of-Thought (CoT) and few-shot prompting, improvements remain marginal. Furthermore, current LLMs struggle to generate high-quality clarifying questions, primarily due to a lack of conflict resolution capabilities and inaccurate utilization of intrinsic knowledge. The second dataset is IN3 [33], a benchmark that evaluates interactive agent capabilities in task-oriented scenarios by assessing explicit task ambiguity resolution and user intent understanding. Although IN3 also releases a training set, we only use its test set for model evaluation and do not incorporate its training data.

### 4.3 Evaluation Metrics

Our evaluation framework incorporates two principal computational metrics to comprehensively assess clarification. First, **Clarification Necessity** is quantified through classification **accuracy (Acc)** and macro **F1-score (F1)**, measuring the model’s capability to discern when user questions require LLMs’ clarification or not. The two metrics evaluate both the precision of binary necessity detection (Acc) and the balanced performance across true positive/negative rates (F1), particularly crucial given the potential class imbalance in ambiguous queries. Second, **Clarifying Question Quality** is evaluated through two complementary measures: (1) **Token F1 (TF)** calculates the percentage of ground-truth clarification points addressed in generated questions, ensuring coverage of essential disambiguation aspects. To calculate TF accurately, we remove all stop-words in the model outputs and only retain important and informative content. (2) **BertScore-F1 (Bert)** assesses semantic alignment between generated and expert-authored clarifications using contextualized BERT embeddings, providing a robust evaluation of linguistic coherence and pragmatic appropriateness. This dual-axis evaluation protocol combines discrete classification performance with nuanced text generation quality metrics, offering a multi-faceted perspective on both the detection and resolution of query ambiguity. We further apply GPT-4 for comparative evaluation [25] simulating human evaluation to tackle the limitation of automated evaluation metrics.

### 4.4 Baseline Models

While existing approaches predominantly focus on domain-specific applications, ClariLM pioneers the investigation of LLM-based clarification mechanisms in general-purpose and open-domain conversational settings. This methodological distinction necessitates comparative evaluation against broadly capable models rather than

<sup>4</sup>Training and test data: <https://huggingface.co/datasets/ZillionZhao/ClariLM>

**Table 2: Main evaluation results of ClariLM and baseline models on three test sets. The best result for each metric is marked in bold and the second best result for each metric is underlined.**

Group	Model	Our-test				IN3				CLAMBER			
		Clari. Necessity		Clari. Quality		Clari. Necessity		Clari. Quality		Clari. Necessity		Clari. Quality	
		Acc	F1	TF	Bert	Acc	F1	TF	Bert	Acc	F1	TF	Bert
LLM	LLaMA-3-8B	73.69	78.12	16.48	78.98	77.88	85.89	23.26	80.12	51.64	58.36	8.30	<b>73.36</b>
	LLaMA-3-70B	78.58	82.76	16.47	81.00	77.78	86.36	23.34	79.21	53.91	57.43	8.80	72.98
	DeepSeek-V3	76.67	78.91	14.57	80.30	72.22	81.71	17.95	74.96	59.94	55.42	7.57	72.14
	GPT-3.5	62.82	76.95	23.27	71.12	87.96	93.60	30.76	85.49	50.33	60.89	<b>10.95</b>	63.16
	GPT-4o	77.60	81.20	19.77	81.99	82.78	86.36	27.02	80.31	59.26	55.11	8.53	72.34
LRM	QwQ-32B	74.83	78.98	13.49	78.38	75.93	84.88	16.67	76.07	61.12	<u>63.79</u>	<u>10.82</u>	71.48
	DeepSeek-R1	79.38	84.00	14.96	79.08	83.33	90.22	18.64	77.95	62.35	63.26	8.54	71.95
SFT	SFT-IN3 [33]	74.16	75.89	15.80	80.81	<b>90.85</b>	94.27	30.68	79.33	57.51	58.88	9.39	70.53
	SFT- $Q^L$ only	79.50	84.28	<u>24.96</u>	83.10	84.26	91.01	29.77	83.70	60.15	57.64	6.50	72.28
	SFT-Full	<u>80.85</u>	<u>84.90</u>	24.71	<u>84.09</u>	<u>90.74</u>	<b>94.74</b>	<b>31.12</b>	<b>87.94</b>	<u>62.53</u>	59.92	7.06	<u>73.27</u>
Our	ClariLM	<b>81.25</b>	<b>85.48</b>	<b>25.19</b>	<b>84.23</b>	89.72	<u>94.36</u>	30.42	<u>86.20</u>	<b>64.23</b>	<b>67.61</b>	9.46	72.32

task-optimized systems. We systematically organize baseline approaches into three principal categories to ensure comprehensive benchmarking. First, **Direct Prompting** evaluates zero-shot capabilities of foundation LLMs: LLaMA-3-7B, LLaMA3-70B, DeepSeek-V3, GPT-3.5-Turbo, and GPT-4o, representing varying scales of pretrained knowledge without clarification-specific tuning. Second, **Reasoning Models** assess explicit logical processing architectures, including QwQ-32B and DeepSeek-R1 (chain-of-thought reasoning model trained with reinforcement learning). Third, **Fine-Tuned Models** examine domain-adapted performance through SFT variants: SFT-IN3 (trained on IN3 [33] training data), SFT- $Q^L$  only (optimized with  $Q^L$  without synthetic data  $Q^G$ ), and SFT-Full (combining log data  $Q^L$ , synthetic data  $Q^G$ , and IN3 training data, but without DPO training). This categorization enables systematic analysis of performance factors across different model scales (7B, 70B, close-source), architectural paradigms (direct generation vs. explicit reasoning), and training strategies (zero-shot vs. fine-tuned approaches), establishing rigorous baselines for general-domain clarification capability assessment.

## 4.5 Implementation Details

We implement both training and inference procedures for FM-LOG and ClariLM using the LLaMA-Factory framework [58], with LLaMA3-8B serving as the base LLM for both architectures. All training experiments are conducted on a computational cluster equipped with four NVIDIA A100-80G GPUs, utilizing PyTorch 2.4.0 and Transformers 4.45.2 frameworks. For the SFT phase, we employ parameter-efficient adaptation through low-rank adaptation (LoRA) [15] with the following configurations: rank dimension 8, initial learning rate of  $1 \times 10^{-4}$ , effective batch size of 32 (physical batch size 8 with 4 gradient accumulation steps), cosine annealing learning rate scheduler, and a single training epoch. During the DPO stage, we adjust hyperparameters to a learning rate of  $1 \times 10^{-5}$ , scaling factor  $\beta=0.1$ , and increase the batch size to 16 to enhance training stability. For the FM-LLM inference and data synthesis in

our OFS framework, we leverage GPT-4o<sup>5</sup> as the foundation model due to its state-of-the-art generation capabilities. Additionally, we employ DeepSeek-R1<sup>6</sup> for reasoning and annotation of the optimal facet during the OFS stage. Both models are implemented through their respective official APIs using the OpenAI client library. Regarding baseline implementations, we directly utilize official APIs for LLM and LRM models. For ClariLM and SFT baselines, we deploy inference on two NVIDIA A100-80G GPUs with generation parameters set to temperature 0.5 and sampling top-p value 0.9, ensuring balanced diversity and coherence in model outputs. All experimental configurations maintain consistency across hardware and software environments to ensure fair comparison.

## 4.6 Experimental Results

**4.6.1 Automated Metrics Evaluation.** As illustrated in Table 2, our main experimental results show that ClariLM achieves state-of-the-art (SOTA) performance across most evaluation metrics across three benchmarks, indicating its exceptional clarification decision-making capability and proficiency in generating high-quality clarifying questions. Notably, despite being an 8B-parameter model, ClariLM outperforms significantly larger LLMs like GPT-3.5 in clarification tasks. The substantial performance gains on our in-domain Our-test benchmark align with expectations given our synthesized training data distribution. More remarkably, ClariLM exhibits superior generalization capabilities, achieving even greater improvements on out-of-distribution benchmarks IN3 and CLAMBER without exposure to their training data.

Specifically, ClariLM demonstrates exceptional performance in judging clarification necessity. Although implicitly determining this during the generation process, ClariLM achieves state-of-the-art results on clarification necessity in both Our-test and CLAMBER benchmarks, while performing comparably to SFT-IN3 (fine-tuned on the IN3 training set) on the IN3 dataset. In contrast, baseline

<sup>5</sup>GPT-4o: <https://chatgpt.com>

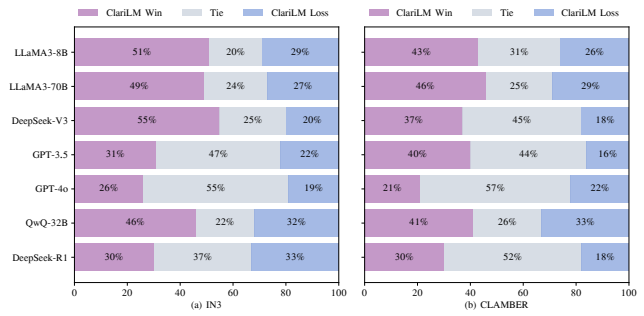
<sup>6</sup>DeepSeek-R1: <https://www.deepseek.com>

models exhibit significant variations and instability in clarification necessity capabilities, which show no strong correlation with their general generation abilities. For instance, GPT-3.5 achieves only 62.82% accuracy on Our-test, substantially lower than other baselines, yet attains 87.96% accuracy on IN3, significantly outperforming other baselines. This indicates that baseline methods lack robust generalization in the clarification task.

Regarding the evaluation of clarification question quality, ClariLM achieves optimal scores on both metrics in Our-test while maintaining competitive performance close to the best results on IN3 and CLAMBER. This confirms ClariLM’s ability to generate high-quality clarification questions. However, lexical and semantic metrics exhibit inherent limitations in quality assessment, as they cannot fully capture the model’s comprehension of user intent [49]. To address this, we further introduce a GPT-4-based simulated human evaluation framework, implementing comparative scoring between pairs of generated candidate responses.

Furthermore, the ablation study on SFT variants also yields three insights: (1) First, SFT-IN3’s domain-specific training reduces cross-domain generalization as it shows sub-optimal performance on Our-test and CLAMBER. (2) The SFT- $Q^L$  variant trained solely on data mined from conversation logs shows performance degradation compared to full data training, emphasizing the importance of our synthetic clarification data. (3) Compared to SFT-Full, our DPO integration provides consistent gains, validating the effectiveness of our preference alignment strategy and negative selection method. These findings collectively confirm that ClariLM’s success stems from synergistic data engineering and preference-based optimization rather than simple parameter scaling.

**4.6.2 GPT-4-based Comparative Evaluation.** Although automated evaluation metrics partially demonstrate ClariLM’s superior clarification decision-making capability and higher-quality question generation compared to baseline methods, they remain insufficient for comprehensive evaluation [25]. To address this limitation, following existing prompting frameworks [25], we employed GPT-4 to conduct comparative assessments between ClariLM and baseline models. We provided GPT-4 with task context and objectives, instructing it to judge the relative quality of outputs using Win, Tie, or Lose evaluations. Considering cost constraints, we evaluated 100 samples from each test dataset. The statistical results shown in Figure 4 reveal that the comparative evaluation trends generally align with the main experimental outcomes in Table 2, both indicating ClariLM’s advantages over baseline methods. Notably, ClariLM demonstrates more pronounced superiority in these human-like evaluations than numerical metrics suggest. For instance, while LLaMA3-8B achieves higher scores on certain metrics in Table 2, it rarely outperforms ClariLM across all test sets in comparative evaluation. Furthermore, GPT-3.5, GPT-4o, and DeepSeek-R1 exhibit comparable performance to ClariLM in many Tie outcomes during comparative evaluation, despite showing a less metric-based advantage in Table 2. This discrepancy suggests that more capable large language models possess greater potential for clarification capabilities than conventional metrics indicate. Besides, although the models exhibit significant variations in performance across different datasets when evaluated using automated metrics, their



**Figure 4: Comparative evaluations between ClariLM vs. seven baseline LLMs on IN3 and CLAMBER benchmarks.**

performance differences become relatively minor in GPT-4o-based comparative evaluations across various test data.

## 5 Conclusion

In this paper, we introduce an LLM-based clarification model ClariLM together with a novel two-stage detection-selection framework for large-scale LLM clarification data synthesizing. By leveraging both conversation logs (FM-LOG) and LLM-guided facet generation (FM-LLM), ClariLM effectively captures realistic and comprehensive clarification facets, enhancing recall in detecting missing user intent. Furthermore, the Optimal Facet Selection (OFS) phase ensures that the model makes informed decisions on whether to ask a clarifying question and, if so, which facet to target, thereby improving precision. The collected data are then applied for SFT followed by DPO training of ClariLM. Through extensive evaluations on three benchmarks together with GPT-4-based comparative evaluation, ClariLM demonstrates superior clarification ability over existing LLMs, effectively balancing when to ask questions and when to respond directly. These results highlight the potential of LLM log-driven learning and preference modeling in general clarification tasks, paving the way for future research on adaptive and domain-agnostic conversational models.

## Acknowledgments

This work was supported by National Science and Technology Major Project No. 2022ZD0120103, Beijing Municipal Science and Technology Project No. Z231100010323009, and National Natural Science Foundation of China No. 62272467. The work was partially done at the Beijing Key Laboratory of Research on Large Models and Intelligent Governance.

## GenAI Usage Disclosure

In this paper, GenAI is primarily used for synthesizing partial data in the methodology, with the relevant details explicitly stated in the main text. Additionally, while GenAI is not employed in drafting the manuscript from scratch, it (GPT-4o) is utilized for error checking (including grammar, tense, etc.) after manual completion.

## References

- [1] Mohammad Aliannejadi, Leif Azzopardi, Hamed Zamani, Evangelos Kanoulas, Paul Thomas, and Nick Craswell. 2021. Analysing Mixed Initiatives and Search Strategies during Conversational Search. *CoRR* abs/2109.05955 (2021). arXiv:2109.05955 <https://arxiv.org/abs/2109.05955>

- [2] Mohammad Aliannejadi, Julia Kiseleva, Aleksandr Chuklin, Jeff Dalton, and Mikhail Burtsev. 2020. ConvA13: Generating Clarifying Questions for Open-Domain Dialogue Systems (ClariQ). *CoRR abs/2009.11352* (2020). arXiv:2009.11352 <https://arxiv.org/abs/2009.11352>
- [3] Mohammad Aliannejadi, Julia Kiseleva, Aleksandr Chuklin, Jeffrey Dalton, and Mikhail Burtsev. 2021. Building and Evaluating Open-Domain Dialogue Corpora with Clarifying Questions. *CoRR abs/2109.05794* (2021). arXiv:2109.05794 <https://arxiv.org/abs/2109.05794>
- [4] Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. 2019. Asking Clarifying Questions in Open-Domain Information-Seeking Conversations. *CoRR abs/1907.06554* (2019). arXiv:1907.06554 <http://arxiv.org/abs/1907.06554>
- [5] Chinmaya Andukuri, Jan-Philipp Fränken, Tobias Gerstenberg, and Noah D. Goodman. 2024. STaR-GATE: Teaching Language Models to Ask Clarifying Questions. *CoRR abs/2403.19154* (2024). arXiv:2403.19154 <https://doi.org/10.48550/arXiv.2403.19154>
- [6] Keping Bi, Qingyao Ai, and W. Bruce Croft. 2021. Asking Clarifying Questions Based on Negative Feedback in Conversational Search. *CoRR abs/2107.05760* (2021). arXiv:2107.05760 <https://arxiv.org/abs/2107.05760>
- [7] Qibin Chen, Junyang Lin, Yichang Zhang, Ming Ding, Yukuo Cen, Hongxia Yang, and Jie Tang. 2019. Towards Knowledge-Based Recommender Dialog System. *CoRR abs/1908.05391* (2019). arXiv:1908.05391 <http://arxiv.org/abs/1908.05391>
- [8] Yang Deng, Yaliang Li, Fei Sun, Bolin Ding, and Wai Lam. 2021. Unified Conversational Recommendation Policy Learning via Graph-based Reinforcement Learning. *CoRR abs/2105.09710* (2021). arXiv:2105.09710 <https://arxiv.org/abs/2105.09710>
- [9] Yihe Deng, Weitong Zhang, Zixiang Chen, and Quanquan Gu. 2023. Rephrase and Respond: Let Large Language Models Ask Better Questions for Themselves. *CoRR abs/2311.04205* (2023). arXiv:2311.04205 <https://doi.org/10.48550/arXiv.2311.04205>
- [10] Luke Friedman, Sameer Ahuja, David Allen, Zhenning Tan, Hakim Sidahmed, Changbo Long, Jun Xie, Gabriel Schubiner, Ajay Patel, Harsh Lara, Brian Chu, Zexi Chen, and Manoj Tiwari. 2023. Leveraging Large Language Models in Conversational Recommender Systems. *CoRR abs/2305.07961* (2023). arXiv:2305.07961 <https://doi.org/10.48550/arXiv.2305.07961>
- [11] Helia Hashemi, Hamed Zamani, and W. Bruce Croft. 2020. Guided Transformer: Leveraging Multiple External Sources for Representation Learning in Conversational Search. *CoRR abs/2006.07548* (2020). arXiv:2006.07548 <https://arxiv.org/abs/2006.07548>
- [12] Helia Hashemi, Hamed Zamani, and W. Bruce Croft. 2021. Learning Multiple Intent Representations for Search Queries. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, Gianluca Demartini, Guido Zuccon, J. Shane Culpepper, Zi Huang, and Hanghang Tong (Eds.). ACM, 669–679. <https://doi.org/10.1145/3459637.3482445>
- [13] Helia Hashemi, Hamed Zamani, and W. Bruce Croft. 2022. Stochastic Optimization of Text Set Generation for Learning Multiple Query Intent Representations. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, Mohammad Al Hasan and Li Xiong (Eds.). ACM, 4003–4008. <https://doi.org/10.1145/3511808.3557666>
- [14] Zhankui He, Zhouhang Xie, Harald Steck, Dawen Liang, Rahul Jha, Nathan Kallus, and Julian McAuley. 2025. Reindex-then-adapt: Improving large language models for conversational recommendation. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, 866–875.
- [15] 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>
- [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] Antonios Minas Krasakis, Mohammad Aliannejadi, Nikos Voskarides, and Evangelos Kanoulas. 2020. Analysing the Effect of Clarifying Questions on Document Ranking in Conversational Search. *CoRR abs/2008.03717* (2020). arXiv:2008.03717 <https://arxiv.org/abs/2008.03717>
- [18] Vaibhav Kumar, Vikas Raunak, and Jamie Callan. 2020. Ranking Clarification Questions via Natural Language Inference. *CoRR abs/2008.07688* (2020). arXiv:2008.07688 <https://arxiv.org/abs/2008.07688>
- [19] Joosung Lee and Jinhong Kim. 2024. Enhanced Facet Generation with LLM Editing. *CoRR abs/2403.16345* (2024). arXiv:2403.16345 <https://doi.org/10.48550/arXiv.2403.16345>
- [20] Wenqiang Lei, Xiangnan He, Yisong Miao, Qingyun Wu, Richang Hong, Min-Yen Kan, and Tat-Seng Chua. 2020. Estimation-Action-Reflection: Towards Deep Interaction Between Conversational and Recommender Systems. *CoRR abs/2002.09102* (2020). arXiv:2002.09102 <https://arxiv.org/abs/2002.09102>
- [21] Wenqiang Lei, Gangyi Zhang, Xiangnan He, Yisong Miao, Xiang Wang, Liang Chen, and Tat-Seng Chua. 2020. Interactive Path Reasoning on Graph for Conversational Recommendation. *CoRR abs/2007.00194* (2020). arXiv:2007.00194 <https://arxiv.org/abs/2007.00194>
- [22] Belinda Z. Li, Alex Tamkin, Noah D. Goodman, and Jacob Andreas. 2023. Eliciting Human Preferences with Language Models. *CoRR abs/2310.11589* (2023). arXiv:2310.11589 <https://doi.org/10.48550/arXiv.2310.11589>
- [23] Wenhan Liu, Ziliang Zhao, Yutao Zhu, and Zhicheng Dou. 2024. Mining Exploratory Queries for Conversational Search. In *Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, May 13-17, 2024*, Tat-Seng Chua, Chong-Wah Ngo, Ravi Kumar, Hady W. Lauw, and Roy Ka-Wei Lee (Eds.). ACM, 1386–1394. <https://doi.org/10.1145/3589334.3645424>
- [24] Zhaocheng Liu, Quan Tu, Wen Ye, Yu Xiao, Zhishou Zhang, Hengfu Cui, Yalun Zhu, Qiang Ju, Shizheng Li, and Jian Xie. 2025. Exploring the Inquiry-Diagnosis Relationship with Advanced Patient Simulators. *arXiv preprint arXiv:2501.09484* (2025).
- [25] Adian Lusia, Potsawee Manakul, and Mark Gales. 2024. LLM Comparative Assessment: Zero-shot NLG Evaluation through Pairwise Comparisons using Large Language Models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, 139–151.
- [26] Yosi Mass, Doron Cohen, Asaf Yehudai, and David Konopnicki. 2021. Conversational Search with Mixed-Initiative - Asking Good Clarification Questions backed-up by Passage Retrieval. *CoRR abs/2112.07308* (2021). arXiv:2112.07308 <https://arxiv.org/abs/2112.07308>
- [27] 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.
- [28] 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. *arXiv preprint arXiv:2410.15576* (2024).
- [29] Fengran Mo, Kelong Mao, Yutao Zhu, Yihong Wu, Kaiyu Huang, and Jian-Yun Nie. 2023. ConvGQR: Generative Query Reformulation for Conversational Search. *CoRR abs/2305.15645* (2023). arXiv:2305.15645 <https://doi.org/10.48550/arXiv.2305.15645>
- [30] 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.
- [31] Fangwen Mu, Lin Shi, Song Wang, Zhuohao Yu, Binquan Zhang, Chenxue Wang, Shichao Liu, and Qing Wang. 2023. ClarifyGPT: Empowering LLM-based Code Generation with Intention Clarification. *CoRR abs/2310.10996* (2023). arXiv:2310.10996 <https://doi.org/10.48550/arXiv.2310.10996>
- [32] Shiyu Ni, Keping Bi, Jiafeng Guo, and Xueqi Cheng. 2023. A Comparative Study of Training Objectives for Clarification Facet Generation. *CoRR abs/2310.00703* (2023). arXiv:2310.00703 <https://doi.org/10.48550/arXiv.2310.00703>
- [33] Cheng Qian, Bingxiang He, Zhong Zhuang, Jia Deng, Yujia Qin, Xin Cong, Zhong Zhang, Jie Zhou, Yankai Lin, Zhiyuan Liu, et al. 2024. Tell Me More! Towards Implicit User Intention Understanding of Language Model Driven Agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1088–1113.
- [34] 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.
- [35] Sudha Rao and Hal Daumé III. 2018. Learning to Ask Good Questions: Ranking Clarification Questions using Neural Expected Value of Perfect Information. *CoRR abs/1805.04655* (2018). arXiv:1805.04655 <http://arxiv.org/abs/1805.04655>
- [36] Sudha Rao and Hal Daumé III. 2019. Answer-based Adversarial Training for Generating Clarification Questions. *CoRR abs/1904.02281* (2019). arXiv:1904.02281 <http://arxiv.org/abs/1904.02281>
- [37] Chris Samarinas, Arkin Dharawat, and Hamed Zamani. 2022. Revisiting Open Domain Query Facet Extraction and Generation. In *ICTIR '22: The 2022 ACM SIGIR International Conference on the Theory of Information Retrieval, Madrid, Spain, July 11 - 12, 2022*, Fabio Crestani, Gabriella Pasi, and Éric Gaussier (Eds.). ACM, 43–50. <https://doi.org/10.1145/3539813.3545138>
- [38] Ivan Sekulic, Mohammad Aliannejadi, and Fabio Crestani. 2021. Towards Facet-Driven Generation of Clarifying Questions for Conversational Search. In *ICTIR '21: The 2021 ACM SIGIR International Conference on the Theory of Information Retrieval, Virtual Event, Canada, July 11, 2021*, Faegheh Hasibi, Yi Fang, and Akiko Aizawa (Eds.). ACM, 167–175. <https://doi.org/10.1145/3471158.3472257>
- [39] Leila Tavakoli. 2020. Generating Clarifying Questions in Conversational Search Systems. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, Mathieu d'Aquin, Stefan Dietze, Claudia Hauff, Edward Curry, and Philippe Cudré-Mauroux (Eds.). ACM, 3253–3256. <https://doi.org/10.1145/3340531.3418513>
- [40] Leila Tavakoli, Hamed Zamani, Falk Scholer, William Bruce Croft, and Mark Sanderson. 2022. Analyzing clarification in asynchronous information-seeking conversations. *J. Assoc. Inf. Sci. Technol.* 73, 3 (2022), 449–471. <https://doi.org/10.48550/arXiv.2007.00194>

- 1002/asi.24562
- [41] Xiaolei Wang, Xinyu Tang, Wayne Xin Zhao, Jingyuan Wang, and Ji-Rong Wen. 2023. Rethinking the Evaluation for Conversational Recommendation in the Era of Large Language Models. *CoRR abs/2305.13112* (2023). arXiv:2305.13112 <https://doi.org/10.48550/arXiv.2305.13112>
- [42] Zhenduo Wang, Yuancheng Tu, Corby Rosset, Nick Craswell, Ming Wu, and Qingyao Ai. 2023. Zero-shot Clarifying Question Generation for Conversational Search. *CoRR abs/2301.12660* (2023). arXiv:2301.12660 <https://doi.org/10.48550/arXiv.2301.12660>
- [43] Zhenduo Wang, Zhichao Xu, Qingyao Ai, and Vivek Srikumar. 2023. An In-depth Investigation of User Response Simulation for Conversational Search. *CoRR abs/2304.07944* (2023). arXiv:2304.07944 <https://doi.org/10.48550/arXiv.2304.07944>
- [44] Julia White, Gabriel Poesia, Robert X. D. Hawkins, Dorsa Sadigh, and Noah D. Goodman. 2021. Open-domain clarification question generation without question examples. *CoRR abs/2110.09779* (2021). arXiv:2110.09779 <https://arxiv.org/abs/2110.09779>
- [45] Cen Yan, Jun Bai, Yanmeng Wang, Wenge Rong, Yuanxin Ouyang, and Zhang Xiong. 2023. Goal-oriented conditional variational autoencoders for proactive and knowledge-aware conversational recommender system. *Comput. Speech Lang. 79* (2023), 101468. <https://doi.org/10.1016/j.csl.2022.101468>
- [46] Lili Yu, Howard Chen, Sida Wang, Yoav Artzi, and Tao Lei. 2019. Interactive Classification by Asking Informative Questions. *CoRR abs/1911.03598* (2019). arXiv:1911.03598 <http://arxiv.org/abs/1911.03598>
- [47] Yifei Yuan, Clemencia Siro, Mohammad Aliannejadi, Maarten de Rijke, and Wai Lam. 2024. Asking Multimodal Clarifying Questions in Mixed-Initiative Conversational Search. *CoRR abs/2402.07742* (2024). arXiv:2402.07742 <https://doi.org/10.48550/arXiv.2402.07742>
- [48] Hamed Zamani, Susan T. Dumais, Nick Craswell, Paul N. Bennett, and Gord Lueck. 2020. Generating Clarifying Questions for Information Retrieval. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, Yennun Huang, Irwin King, Tie-Yan Liu, and Maarten van Steen (Eds.). ACM / IW3C2, 418–428. doi:10.1145/3366423.3380126
- [49] Hamed Zamani, Gord Lueck, Everest Chen, Rodolfo Quispe, Flint Luu, and Nick Craswell. 2020. MIMICS: A Large-Scale Data Collection for Search Clarification. *CoRR abs/2006.10174* (2020). arXiv:2006.10174 <https://arxiv.org/abs/2006.10174>
- [50] Hamed Zamani, Bhaskar Mitra, Everest Chen, Gord Lueck, Fernando Diaz, Paul N. Bennett, Nick Craswell, and Susan T. Dumais. 2020. Analyzing and Learning from User Interactions for Search Clarification. *CoRR abs/2006.00166* (2020). arXiv:2006.00166 <https://arxiv.org/abs/2006.00166>
- [51] Michael JQ Zhang, W Bradley Knox, and Eunsol Choi. 2024. Modeling future conversation turns to teach llms to ask clarifying questions. *arXiv preprint arXiv:2410.13788* (2024).
- [52] Tong Zhang, Peixin Qin, Yang Deng, Chen Huang, Wenqiang Lei, Junhong Liu, Dingnan Jin, Hongru Liang, and Tat-Seng Chua. 2024. CLAMBER: A benchmark of identifying and clarifying ambiguous information needs in large language models. *arXiv preprint arXiv:2405.12063* (2024).
- [53] Yiming Zhang, Lingfei Wu, Qi Shen, Yitong Pang, Zhihua Wei, Fangli Xu, Bo Long, and Jian Pei. 2022. Multiple Choice Questions based Multi-Interest Policy Learning for Conversational Recommendation. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, Frédérique Laforest, Raphaël Troncy, Elena Simperl, Deepak Agarwal, Aristides Gionis, Ivan Herman, and Lionel Médini (Eds.). ACM, 2153–2162. <https://doi.org/10.1145/3485447.3512088>
- [54] Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024. Wildchat: 1m chatgpt interaction logs in the wild. *arXiv preprint arXiv:2405.01470* (2024).
- [55] Ziliang Zhao and Zhicheng Dou. 2024. Generating Multi-turn Clarification for Web Information Seeking. In *Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, May 13-17, 2024*, Tat-Seng Chua, Chong-Wah Ngo, Ravi Kumar, Hady W. Lauw, and Roy Ka-Wei Lee (Eds.). ACM, 1539–1548. <https://doi.org/10.1145/3589334.3645712>
- [56] 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, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, Ambuj K. Singh, Yizhou Sun, Leman Akoglu, Dimitrios Gunopulos, Xifeng Yan, Ravi Kumar, Fatma Ozcan, and Jieping Ye (Eds.). ACM, 3549–3558. <https://doi.org/10.1145/3580305.3599389>
- [57] Ziliang Zhao, Zhicheng Dou, Jiabin Mao, and Ji-Rong Wen. 2022. Generating Clarifying Questions with Web Search Results. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, Enrique Amigó, Pablo Castells, Julio Gonzalo, Ben Carterette, J. Shane Culpepper, and Gabriella Kazai (Eds.). ACM, 234–244. <https://doi.org/10.1145/3477495.3531981>
- [58] Yaowei Zheng, Richong Zhang, Junhao Zhang, YeYanhan YeYanhan, and Zheyang 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.
- [59] Kun Zhou, Wayne Xin Zhao, Shuqing Bian, Yuanhang Zhou, Ji-Rong Wen, and Jingsong Yu. 2020. Improving Conversational Recommender Systems via Knowledge Graph based Semantic Fusion. *CoRR abs/2007.04032* (2020). arXiv:2007.04032 <https://arxiv.org/abs/2007.04032>
- [60] Kun Zhou, Yuanhang Zhou, Wayne Xin Zhao, Xiaoke Wang, and Ji-Rong Wen. 2020. Towards Topic-Guided Conversational Recommender System. *CoRR abs/2010.04125* (2020). arXiv:2010.04125 <https://arxiv.org/abs/2010.04125>