Table 1: Proposed action-based ambiguity type.

Ambiguity Type	Definition	Related ATs from previous work
Semantic	The query is semantically ambiguous for several common reasons: it may include homonyms; a word in the query may refer to a specific entity while also functioning as a common word; or an entity mentioned in the query could refer to multiple distinct entities.	*Question Focus* [35] *Linguistic Ambiguity* [61]
Generalize	The query focuses on specific information; however, a broader, closely related query might better capture the user's true information needs.	*Generalization* [8, 31]
Specify	The query has a clear focus but may encompass too broad a research scope. It is possible to further narrow down this scope by providing more specific information related to the query.	*Faceted* [15] *Time Dependency* [43] *Underspecified References* [3]

under this category (e.g. *NeedsElaboration*, *UnderspecifiedReferences* [3]).

Table 1 presents detailed explanations of our AT taxonomy. Compared to previous taxonomies, the strength of our taxonomy lies in its dual function: each AT in our taxonomy not only helps LLMs understand underlying ambiguities, but can also be easily interpreted as an action for LLMs to take.

3.2 Prompting Formulation for Clarification Generation

This section aims to respond to RQ2, i.e. how to integrate ambiguities, abstracted by the ambiguity type taxonomy in Section 3.1, into reasoning for LLM prompting methods. We aim to achieve this by constraining the reasoning of CoT prompting. Intuitively, we seek to require LLMs to predict ATs in our taxonomy to integrate ambiguities into reasoning, which endows LLMs the capability to reason in a way that we expect and take explainable actions to clarify ambiguous queries. We hypothesize that ambiguity-oriented reasoning is better than freely generated LLM reasoning. Therefore, we propose Ambiguity Type-Chain of Thought (AT-CoT) that extends CoT prompting. To effectively access the impact of integrating ambiguities into LLM reasoning, we use another two prompting schemes as baselines: standard prompting, which simply requires LLMs to generate clarifications without any intermediate steps; AT-standard prompting, for which we add definitions of ATs in our taxonomy into prompt instructions. We use AT-standard to validate the impact of simply informing LLMs of possible ATs without asking LLMs to generate reasoning. Table 2 shows detailed system instructions for different prompting schemes. Mathematically, each prompting method can be formulated as follows:

• **standard** [24]: Standard prompting relies only on inherent knowledge of LLMs to generate clarifications without intermediate steps. The objective of standard prompting is to maximize:

$$p(c|\mathcal{D}, C, q) \tag{1}$$

where c denotes the generated clarification, q denotes an ambiguous query, C denotes the conversation history, and $\mathcal{D}$ denotes the task description.

• **AT-standard**: The only difference between AT-standard and standard prompting is that AT definitions are included in the prompt:

$$p(c|\mathcal{D}, \mathcal{A}, C, q) \tag{2}$$

where $\mathcal{A}$ refers to the AT definitions from Table 1.

• **CoT (Chain of Thought)** [57]: CoT prompting requires LLMs to generate texts of reasoning before making clarifications (reasoning without constraints):

$$p(a, c|\mathcal{D}, C, q) \tag{3}$$

where a refers to the generated textual reasoning.

• **AT-CoT**: AT-CoT requires LLMs to first predict ATs from our taxonomy, then generate clarifications correspondingly. The objective of AT-CoT prompting is to maximize:

$$p(a, c|\mathcal{D}, \mathcal{A}, C, q) \tag{4}$$

4 Experimental Setup

To evaluate the effectiveness of AT-CoT, we conduct experiments on three types of tasks: (1) Clarification generation (CG), for which we use datasets containing human-annotated clarifying questions (CQs) and we evaluate by computing the semantic similarity between generated CQs and human-annotated ones. (2) Information retrieval (IR), for which we simulate multi-turn conversations and transform conversations into reformulated queries to retrieve documents. (3) CG+IR, for which we align CG performance and IR performance to investigate the correlation between the performance of CG and IR, i.e., if better clarifications could improve IR performance.

4.1 Datasets

We present datasets that are used in our experiments in this section. Table 3 summarizes the statistics of different datasets.

4.1.1 CG Datasets.

• Qulac [2]: Qulac uses queries from TREC web track 2009-2012. Annotators are asked to first figure out facets related to given queries by scanning snippets of web searching results using a search engine, then generate CQs to address the facets.

• ClariQ [1]: Similarly to Qulac, ClariQ is crowdsourced by annotating CQs for provided queries. Ambiguity level labels ranging from 1-4 are provided in ClariQ, with 4 representing extreme ambiguous queries.

• RaoCQ [51]: A domain-specific dataset containing clarifying

Table 2: Prompts of four prompting schemes: standard, AT-standard, CoT and AT-CoT. <AT definitions> is a placeholder for AT definitions in Table 1.

Prompt Type	System Instruction
standard	`Given a query in an information-seeking system, generate a clarifying question that you think is most appropriate to gain a better understanding of the user's intent.` `<query>`

AT-standard	`Given a query in an information-seeking system, generate a clarifying question that you think is most appropriate to gain a better understanding of the user's intent. The ambiguity of a query can be multifaceted, and there are multiple possible ambiguity types:`
	`<AT definitions>`
	`Consider the above ambiguity types when generating.`
	`<query>`

CoT	`Given a query in an information-seeking system, generate a clarifying question that you think is most appropriate to gain a better understanding of the user's intent.`
	`Before generating the clarifying question, provide a textual explanation of your reasoning about why the original query is ambiguous and how you plan to clarify it.`
	`<query>`

AT-CoT	`Given a query in an information-seeking system, generate a clarifying question that you think is most appropriate to gain a better understanding of the user's intent. The ambiguity of a query can be multifaceted, and there are multiple possible ambiguity types:`
	`<AT definitions>`
	`Before generating the clarifying question, provide a textual explanation of your reasoning about which types of ambiguity apply to the given query. Based on these ambiguity types, describe how you plan to clarify the original query.`
	`<query>`

question annotations. Annotators are asked to identify relevant CQs given (question, follow-up questions) pairs, where each question refers to an original question from a post on StackExchange, and follow-up questions are sampled from follow-up questions in comments of the same post. Similarly to [51], in this work, we evaluate our methods on the subset with human annotations.

4.1.2 IR Datasets.

• TREC Web track 2009-2012 [15–18]: An IR dataset that focus on web search queries. We use ClueWeb09[1] Category B as the document collection which contains 50 million English web pages. Since facet-specific document relevance judgments are provided in TREC web track diversity tasks, we use facets as user intents in user simulation.

• TREC Web track 2013-2014 [19, 20]: As TREC Web track 2009-2012, TREC Web track 2013-2014 contains multifaceted web search queries, while including more focused topics to present more challenging queries. ClueWeb12[2] is used as the document collection.

• TREC DL Hard [40]: A benchmark containing with queries from TREC DL 2019 & 2020 [21, 22], which we believe may require multi-turn clarification to resolve implied ambiguities. The queries in TREC DL Hard are sampled from MS Marco [10]. We use the MS Marco passage corpus as the document collection.

[0]https://lucene.apache.org/

[1]https://lemurproject.org/clueweb09.php/

[2]https://lemurproject.org/clueweb12/

Table 3: Statistics of datasets used in our experiments for three tasks: CG, IR, CG+IR.

Dataset	# queries	# CQs	# intents
Task 1: CG			
Qulac	198	2575	-
ClariQ	298	3991	-
RaoCQ	500	2248	-
Task 2: IR			
TREC Web Track 2009-2012	198	-	717
TREC Web Track 2013-2014	100	-	315
TREC DL Hard	50	-	350
Task 3: CG+IR			
Qulac-TREC Web Track 2009-2012	198	2575	717

4.1.3 CG+IR Dataset. Since Qulac is based on queries from TREC Web Track 2009-2012, we align Qulac queries to document relevance judgments provided by TREC Web Track 2009-2012. We refer to this dataset by Qulac-TREC Web Track 2009-2012, which contains human-annotated CQs from Qulac and document relevance judgments from TREC Web Track 2009-2012.

4.2 Evaluation Protocol

Clarification Generation. For each query, we generate multiple CQs to fairly evaluate the performance of different prompting methods on the clarification generation (CG) task. Several factors drive

this decision: 1) In CG datasets, each query corresponds to numerous human-annotated CQs, covering different clarification possibilities. However, human-annotated CQs cannot contain all possible CQs. Since automatic metrics such as BERTScore [60] capture the semantic similarity between generated CQs and reference CQs, it is likely that a high-quality generated CQ gets a low BERTScore. To mitigate this issue, we seek to generate multiple diverse CQs, therefore reducing the probability that none of the generated CQs is semantically similar to any of the human-annotated ones. 2) For AT-CoT, since multiple ambiguity types may exist for a query, it is natural to generate multiple CQs to account for different ATs. For other prompting methods that do not predict ATs, generating multiple CQs is also helpful for fair comparison. As evidenced in Wang et al. [54], the voting strategy can increase the performance of LLM prompting methods. Generating multiple CQs and comparing the best-performing CQ can be regarded as a type of voting, by which we reduce the variance of prompting performances on CG.

User Simulation for IR. We also evaluate the impact of our methodology on IR via user simulation. Two clarification interaction scenarios are tested (Figure 3):

• *select*: In each turn, 5 RQs are generated. We adopted a moderate temperature of 0.6 to balance the diversity and consistency of the generated RQs, ensuring that they are varied while avoiding excessive creativity. The user agent selects the RQ that best corresponds to their intent, and the conversation continues based on the selected RQ.

• *respond*: In each turn, a CQ is generated. The user agent responds to it based on the provided user intent.

We choose the two interaction scenarios for the following reasons: 1) *select* is widely studied in previous studies [11, 23], and applied in certain real-world scenarios such as search engine suggestions through query reformulation. 2) *respond* corresponds to the more naturalistic settings for conversational search modeling interactions in natural language [2, 37, 52]. A baseline w/o clarification is also considered, which uses original queries without clarification for document retrieval.

To simulate multi-turn conversations, three prompts are chained: *generation*, *response*, *reformulation*. The *generation* prompt has four variants, each representing a prompting methods as described in Section 3.2. We use this prompt to generate RQs (scenario *select*) or CQs (scenario *respond*). The *response* prompt generates simulated user responses and has two variants, each corresponding to an interaction scenario. The clarifications generated by *generation* and paragraphs describing user intents are used as input for the *response* prompt. Since facet-specific document relevance judgments are provided in TREC Web track 2009-2014, we directly use facets as user intents for *response* and use gold document relevance labels. For TREC DL Hard, we use each relevant document as user intent for *response*, then use the same document as the gold label in IR evaluation. For each simulated conversation, the *reformulation* prompt uses the simulated conversation as input and summarizes the conversation into a reformulated query. The objective of using *reformulation* is to facilitate the evaluation of IR tasks, since most existing IR models retrieve documents by queries rather than conversations. Examples of the *response* and *reformulation* prompts can be found in Table 4. As a result, we have eight types of simulated conversations, each corresponding to a unique (clarification generation prompt, interaction scenario) combination. To simplify the simulation, we assume that users are always cooperative and that no intent shifts occur during the conversation. Each conversation is initialized by a user query and simulated to three turns with no stopping rules. User agents are always provided with complete descriptions of user intents.

4.3 Evaluation Metric

Following [61], we use BERTScore [60] for CG tasks, since metrics based on N-gram matching like BLEU or ROUGE cannot measure clarification abilities [30]. As mentioned in Section 4.2, we ask LLMs to generate multiple CQs. For each query, we compute a score for generated CQs as follows: suppose that for each query q, we generate a list of CQs $(gcq_1, ..., gcq_M)$, with a list of annotated CQs $(acq_1, ..., acq_N)$ as gold standards. We first compute a query-specific score matrix S:

$$S_{i,j} = BERTScore(gcq_i, acq_j) \tag{5}$$

where $i = 1, ..., M$, $j = 1, ..., N$. The score on q is computed by: $score_q = max(S_{[:,:]})$, which is the maximum value of S. We take the BERTScore of the best-performing generated CQ to access the overall CG performance on q for the following reason: a CG method is good if it is able to generate a CQ that is highly similar to one of the reference CQs.

For the IR task, we use different standard metrics following previous work [15, 62]: We use nDCG@10 (Normalized Discounted Cumulative Gain [55]) for TREC web track 2009-2014. For TREC DL Hard, since we use each relevant document as user intent for simulation then verify whether the target document is ranked higher through clarification, we use MRR@10 (Mean Reciprocal Rank [48]) as the evaluation metric.

4.4 Implementation Details

Prompting Scheme. Following previous work [24], we adopt few-shot settings for all prompting schemes. We have two reasons to do so: 1) results of preliminary experiments demonstrate that few-shot prompting always significantly outperform zero-shot, regardless of the prompting scheme; 2) zero-shot prompting is likely to generate over lengthy analysis, causing incomplete generation due to maximum output token limitation or slowing down inference. In this work, we assume without further notice that all prompting methods are under few-shot settings.

LLM. We use Llama-3-8B [25] as our base model and load pretrained weights from Huggingface. LLM hyperparameters are fixed with: $k = 10$ for top-k sampling; temperature $t = 0.6$. Due to the extensive prompting inference involved in our experiments, we quantize Llama-3 to NF4 (4-bit NormalFloat) and conduct our experiments on a single 12G TITAN Xp GPU.

Parsing LLM Outputs. We ask LLMs to give JSON-style structured outputs through format instructions and few-shot examples containing reference formatted outputs. LLM outputs are parsed using the Pydantic parser from LangChain[3]. In case of parsing errors,

[3]https://python.langchain.com/v0.1/docs/modules/model_io/output_parsers/types/pydantic/