

A First Step towards Dialog Simulation with Grounded Dialog Graphs

*Michael Ginn¹ *Matt Pauk¹ Tava Reese¹
Sameer Gupta² Giuseppe Castelvici² Kevin Small² Alessandro Moschitti²
Derek Palmer¹ Martha Palmer¹ Alexis Palmer¹ Maria Leonor Pacheco¹
¹University of Colorado Boulder ²Amazon
{michael.ginn, matt.pauk}@colorado.edu

Abstract

In this work, we propose a method for dialog simulation to gather **high-quality open-domain, multi-turn question answering conversations**. The simulation is grounded on Stack Exchange posts and motivated by computational discourse theory. We first convert forum posts into structured directed graphs; then, different traversals through the graph represent possible conversational trajectories. Our proposed graph traversal algorithm produces dialogs optimized for *conversational efficiency*. In addition, we propose an evaluation framework based on Gricean conversational maxims. Expert-level human annotators evaluate 105 cooking domain transcripts according to our framework; dialogs produced by our method receive ratings that are competitive with dialogs from prior work.

1 Introduction

Large Language Models (LLMs) are highly fluent and accurate conversational question answering systems, long since saturating benchmarks such as CoQA (Reddy et al., 2019). Despite this, there is a scarcity of real-world, publicly available transcripts from information-seeking conversations between users and AI assistants (Soudani et al., 2024). Also, actual human-assistant dialogs tend to be fragmented, incoherent, or overinformative, failing to meet the quality standard needed to train better conversational agents. In lieu of real-world conversations, research has studied methods for **synthetic data generation** of conversational transcripts, taking approaches like transforming textual sources into dialogs (Dai et al., 2022; Bao et al., 2023; Zheng et al., 2023a) or using specialized user-simulation models to generate dialogs turn-by-turn (Mohapatra et al., 2021; Wu et al., 2022; Lin et al., 2022). Most such research targets *domain-specific, task-oriented dialogs* with explicit goals, and there

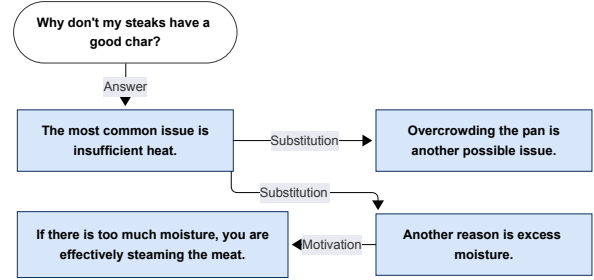


Figure 1: A basic dialog graph. The white pill represents the main question, while blue boxes are parts of the answer. Edges represent information precedence and dialog/discourse relations.

is relatively little work on *general information-seeking dialogs* that may include multiple distinct goals, open-ended exploratory questions, and clarifying questions. Furthermore, we find that naïve approaches (i.e., zero-shot prompting an LLM) for dialog generation tend to suffer from unnatural, unhelpful questions (from either user or assistant) and answers (examples in §2.1).

To solve these issues, we propose a dialog simulation method inspired by theories of discourse (see §2.2). It produces realistic transcripts of general domain, information-seeking conversations, and constrains the simulated AI assistant to respond efficiently, rather than providing near-encyclopedic answers. We use an LLM to apply structural rules that convert Stack Exchange posts into **directed acyclic graphs** (DAGs) which encode many possible conversational trajectories for the same information exchange (see Fig. 2). Our graphs represent information precedence (moving from general to specific claims) motivated by theories of *common ground* (Stalnaker, 2002) and *question under discussion* (Roberts, 2012; Ko et al., 2023). Our graph representations provide a principled framework flexible enough to model various types of information-seeking dialogs; including single-turn question/answer (QA) pairs and multi-turn dialogs

*Equal contribution.

with bidirectional information sharing.¹

We also describe a **traversal algorithm** (executed by an LLM) to generate a dialog from a particular path in the graph, including user *follow-up questions* and *clarifying questions* from the assistant. The algorithm is designed around *Gricean conversational maxims*, principles that describe how humans communicate efficiently (Grice, 1975). Most importantly, we aim to produce dialogs where the AI assistant follows the *maxim of quantity*, which states that speakers should provide as much information as needed, but no more.

In conjunction, we create an **evaluation rubric** based on all four Gricean maxims. The *maxim of relation* calls for contributions to be relevant to the conversation; the *maxim of quality* highlights information accuracy; and the *maxim of manner* states that contributions should be clear and straightforward. Using this rubric, 41 expert-level human annotators blindly annotate 105 dialogs generated with our simulation strategy and 32 generated dialogs from prior work. On average, annotators prefer our dialogs across key dimensions, including enjoyment, efficiency and flow. We also test whether an LLM-as-judge can perform automatic evaluation with our rubric, finding high agreement with human annotators across most metrics of our evaluation framework. However, the LLM struggles with certain metrics that evaluate comprehensiveness of assistant answers and relevance/necessity of claims provided within assistant responses. Finally, we outline a specific plan for future work to scale up our annotated dataset and use it for LLM post-training.

2 Background and Related Work

2.1 Dialog Simulation

Relatively little prior work on dialog simulation aims to simulate general-domain information-seeking dialogs. The standard approaches include crowdworker simulation, turn-by-turn simulation, and transformation of existing texts.

Crowdworker Simulation. One straightforward option is to use crowdsourced human annotation to manually create dialogs, often with a pair of annotators simulating the roles of user and assistant. Dinan et al. (2019) use the "Wizard-of-

Oz" paradigm for crowdsourced dialog simulation, where the "Wizard" has privileged access to a knowledge source (here, Wikipedia) used to answer information-seeking questions. Other studies also leverage Wikipedia as a knowledge source (Choi et al., 2018; Adlakha et al., 2022), as well as Internet search (Komeili et al., 2022), multimodal databases (Liao et al., 2021), and LLM-based RAG (Katsis et al., 2025); still other work extends the paradigm to bidirectional knowledge transfer (Gopalakrishnan et al., 2019). However, human annotation can be expensive to scale, and crowdsourced dialogs often contain errors (Dziri et al., 2022). For example, in the MTRAG dataset (Katsis et al., 2025), many simulated answers demonstrate undesirable behavior for an AI agent, such as the missing word in the user turn below:

User	Turn 5
It is better never be a co-signer.	
Assistant	Turn 6
Actually being a cosigner for a loan involves significant responsibilities. You're just as responsible for the loan payments as the original borrower. Co-signing a loan is just plain stupid.	

Instead, we focus on designing a fully-artificial—and therefore easily scalable—dialog simulation method that facilitates interventions to control desired properties of the conversation.

Agent Simulation. Similar to the crowdworker paradigm, other work simulates dialogs turn-by-turn using one or two AI agents running in a loop (Mohapatra et al., 2021; Wu et al., 2022). While this approach can model realistic conversational dynamics, it is difficult to control the overall information trajectory of the conversation. To do so usually requires comprehensively specifying goals and subgoals for the user, which can be difficult for open-ended knowledge seeking goals.

Direct Generation. One alternative is to generate dialogs directly from existing text such as Wikipedia articles (Dai et al., 2022) or real-world conversational transcripts (Zheng et al., 2023b), using a trained specialized model (Li et al., 2020; Lin et al., 2022) or a pretrained LLM (Mo et al., 2024). This requires no specification of user goals, has a lower risk of hallucinated information, and enables the LLM to model the entire conversation at once. However, conversations generated by these approaches are often formulaic or unnatural, such

¹In this study we focus on multi-turn dialogs in the general domain of cooking. We have performed small proof-of-concept studies for other domains and will study them more systematically in future work.

as this excerpt from Dai et al. (2022):

User	Turn 3
What is important about the village?	
Assistant	Turn 4
At the 2006 census, its population was 281, in 68 families.	

Grounded Agents. Our method unifies these two approaches, providing the grounding and scalability of generation-from-text with the naturalness of simulation, while maintaining a conversational trajectory. We generate conversations turn-by-turn, constraining the generating agent to predefined conversational trajectories encoded in a dialog graph, which is grounded in existing text from Stack Exchange posts. Our work is similar to Maharana et al. (2024), which grounds agents on event graphs, to Gritta et al. (2021); Ouyang et al. (2021) which use *conversational graphs* to create task-based dialogs, and in particular to Bao et al. (2023), which breaks Wikipedia articles into *knowledge pieces* and arranges them by emulating *dialog flows* of real-world conversations. However, they rely on task-specific heuristics to categorize dialog flows, while our method is scalable and needs no task- or domain-specific information.

2.2 Dialogs and Discourse Theory

We ground our approaches to dialog graphs (§3), generation (§4), and evaluation (§5) in theories of discourse and dialog. Specifically, we label graph edges with Dependency Dialogue Act (DDA, Cai et al., 2025) relations; graph nodes are ordered to reflect questions under discussion, information precedence, and common ground; and Gricean maxims guide both graph traversal and dialog evaluation.

Gricean theory (Grice, 1975) provides principles for *conversational implicature*, a pragmatic process that enables successful communication, especially when intended meaning does not align with literal surface form. Gricean framing has been used within NLP to several ends (Krause and Vossen, 2024), including assessing conversational implicature (Zheng et al., 2021; Li et al., 2023; Anurajana et al., 2024; Sravanthi et al., 2024, inter alia), generating dialogs (e.g., Harabagiu et al., 2004), extracting inferences from raw text (Harabagiu and Moldovan, 1996), and structuring communication about AI safety (Kaas and Habli, 2025).

Our work proposes a method for dialog evaluation based on Gricean framing. Prior work has ex-

plored Gricean-motivated evaluation frameworks—for human evaluators and automated systems—of different dialog aspects such as question-answer pairs (Qwaider et al., 2017; Freihat et al., 2018) and follow-up questions (Yadav et al., 2025). Our work is most similar to Jwalapuram (2017); Xiao et al. (2020); Panfili et al. (2021); Miehl et al. (2024); Wölfel et al. (2024), who use LLMs or human annotators to produce dialog-level ratings for each maxim. We do fine-grained evaluation with concrete dialog-, turn- and claim-level metrics.

3 Grounded Dialog Graphs

3.1 Graph Representation

An information-seeking dialog represents a specific trajectory of information exchange, where both participants take turns sharing information and/or asking questions. Often, there are many possible conversational trajectories corresponding to the same set of facts, which may vary in length, number of turns, and style. We represent the full space of possible trajectories using *directed acyclic graphs* (DAGs), as shown in Figure 1. Our graphs bear similarities to the ConvGraphs of Gritta et al. (2021), however, their method focuses on turn-level dialog acts for task-oriented dialogs, rather than statement-level structure in information-seeking dialogs. As most dialog turns are multi-statement, this makes our approach more fine-grained.

Nodes. Nodes in our graph correspond to atomic statements (including questions). A single turn may consist of multiple nodes. These statements roughly correspond to *elementary discourse units* (EDUs) as introduced by Rhetorical Structure Theory (Mann and Thompson, 1988; Carlson et al., 2001). Nodes are distinguished as originating from the question asker (user) or the answerer (assistant) with different colors.

Edges. Edges represent **information precedence**; that is, an edge from node A to node B means that node A must have been introduced to the common ground prior to node B. In our running example, the node stating "Another reason is excess moisture" must be introduced prior to "If there is too much moisture, you are effectively steaming the meat.", or the latter turn would be infelicitous. Edges are labeled with rhetorical relations that describe the relationship between the two statements. We use a subset of the Dependency Dialog Acts label inventory (Cai et al., 2025),

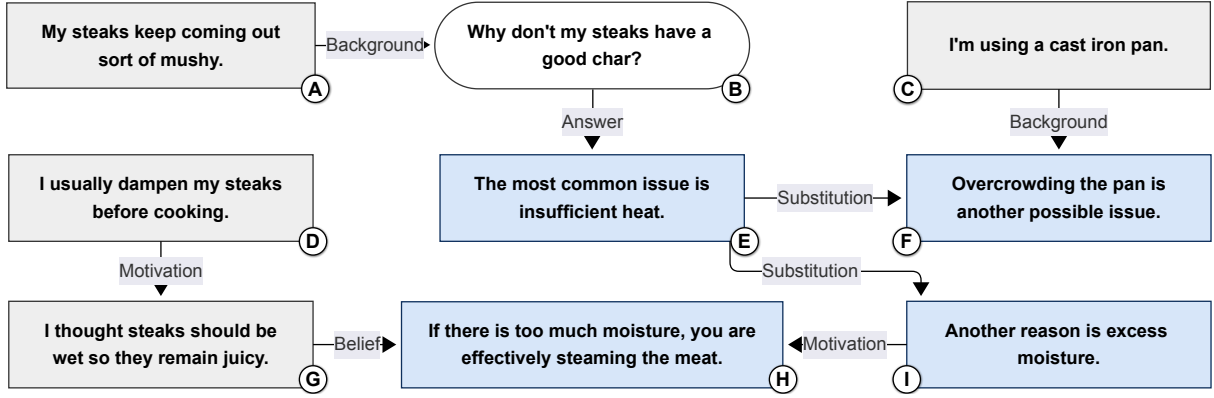


Figure 2: The graph from Figure 1, with added Background and Belief nodes (light grey boxes).

which itself selects from Dialog Act labels (Stolcke et al., 2000) and Penn Discourse Treebank v3 labels (Prasad et al., 2018). Unlike DDA, which allows edges pointing in either direction with respect to discourse flow, our edges are constrained by information precedence and should always point from more general to more specific information. We also add three structural labels: Answer, Background, and Belief, described in the next section. Our inventory appears in App. A.

Graph Structure. A minimal graph (as in Figure 1) consists of a question node and one or more answer nodes. The question node has a single child answer node (connected with an edge labeled Answer) that represents the most critical part of the answer. In addition, information-seeking dialogs often include statements in which the question asker provides information to the answerer. We identify two main categories of these sorts of statements. **Background statements** provide concrete information about the user’s situation or history that is relevant to the question under discussion. **Belief statements** describe the user’s preconceived notions, feelings, or guesses about the question under discussion. We create nodes for background and belief statements and connect related nodes to each other according to the information precedence relation defined earlier. Then, background/belief nodes are connected to the existing graph (either the answer nodes or main question) with edges labeled Background or Belief, indicating that a given parent statement should be provided before the question or answer.

In Figure 2, the statements “I usually dampen my steaks before cooking.” and “I’m using a cast iron pan.” are concrete background statements about the user’s situation. Also, the latter statement is

prerequisite background for the answer about overcrowding the pan—if the asker was using a grill rather than a pan, this answer would not make sense. Although the other statement (“I usually...”) is a Background node, it is connected to the graph via a Belief edge. This describes the user’s beliefs about adding moisture to a steak.

The graph superimposes many possible conversational trajectories and serves as a unified representation for different forms of information-seeking dialogs. In §4, we describe a procedure that uses graph intermediaries to convert single-turn QA posts into multi-turn, bidirectional dialogs.

3.2 Automated Graph Construction

The first step of our dialog generation method is to convert existing posts from Stack Exchange (a question answering forum with topic-specific subsites: <https://stackexchange.com>) into dialog graphs. Stack Exchange posts are single-turn interactions with a single question and one or more answers. For a given Stack Exchange subsite, we include only questions with high-consensus answers (in the top 90th percentile of votes). In addition, we filter to include only posts where the top answer has at least six sentences to ensure that there is enough information to work with, and we always choose the top-rated answer. Figure 4 in Appendix C shows an example question/answer pair.

Graphs are created as follows: 1) split the question and answer into a list of statements with nodes for each statement; 2) prompt an LLM to build a graph given the question node and answer nodes (but not background/belief nodes); and 3) prompt an LLM to add the background and belief nodes to the existing graph. All steps use GPT 5.2.

For step one, we use two separate LLM prompts

Node Type	Description
Question	The main question raised in the post, or clarifying questions raised in the answer.
Subquestion	An additional question raised by the asker related to the main question.
Claim	A verifiable, widely-known fact that addresses the question under discussion.
Anecdote	A personal story from the answerer.
Meta	A metatextual statement that references the question itself or the forum site.
Belief	A statement describing beliefs or preferences about the question under discussion.
Background	A concrete fact about the user’s current situation that provides context for the question.

Table 1: Node types in the dialog graph.

(see [Appendix B](#)) for the question and answer, with a single handcrafted in-context example for all dialogs. The LLM then classifies each statement into one of the node types from [Table 1](#). We immediately filter out any Meta and Anecdote statements, since these generally do not fit human-AI conversations.² We also filter out Subquestions.

Next, we prompt the LLM to structure the graph by predicting edges between the main Question statement and all statements that make up the answer (see [Appendix B](#)). We provide detailed rules about the structure of the graph, a list of available edge labels, and a handcrafted in-context example. We request the list of edges to be returned in the Mermaid chart language, and the LLM always conformed to the expected format.³ Because there are often multiple possible structures for a given graph, it is not crucial to rigorously validate the predicted structures. During development, we qualitatively evaluated the generated graphs (see [Appendix C](#)) and found that they generally adhered to our expectations. Finally, we add background and belief nodes from the original question to the graph with another LLM prompt. We provide the full graph structure and the new nodes to be added, and ask the LLM to 1) connect background nodes to one another using DDA-labeled edges; and 2) connect background nodes to existing nodes on the graph (following the rules described in §3.1). Full graphs are saved as intermediate representations, which we now use to generate dialog transcripts.

²We initially tried to filter these in the decomposition prompt, but found that this did not work well.

³<https://github.com/mermaid-js/mermaid>

4 Dialog Generation

Algorithm. A dialog is generated by selecting one particular trajectory through the full graph. Any trajectory through the graph will produce a pragmatically well-formed conversation, so long as the precedence relation (i.e. the direction of each edge) is followed. For example, a simple pre-order traversal of the entire answer subtree would produce a one-turn dialog very similar to the original Stack Exchange post. This is not, however, a very interesting dialog, so we propose a traversal algorithm motivated by the Gricean maxim of quantity. This produces multi-turn conversations, including follow-up questions from the user and clarifying questions from the assistant. The general algorithm is as follows, with additional details and prompts in [Appendix D](#).

1. Start at the main question node, which becomes the question under discussion (QUD).
2. Generate a user turn for the QUD, including required background/beliefs.
3. Select the answer subtree that is necessary to answer the QUD.
4. If any of the selected answer nodes are downstream of unvisited background nodes, generate a clarifying question to prompt for this background and for the corresponding user answer.
5. Rewrite answer subtree from 3 as fluent answer.
6. Select an unvisited edge outgoing from the full set of visited answer nodes. Generate a follow-up question based on this edge, which becomes the new QUD.
7. Repeat steps 2-6 until there are no remaining nodes or the turn limit is reached.

Example Traversal. Next, we walk through these steps in detail for the graph from [Figure 2](#), resulting in the actual GPT 5.2 outputs in [Figure 3](#).

Turn 1: Main Question. We select the main question node (B) and any upstream background or belief nodes (A).⁴ Then, we use an LLM to rewrite this list of statements into a coherent question.

Turn 2: Answer. Next, we must select the subtree of answer nodes that are sufficient and necessary to answer the QUD (the main question). We always include the top-level answer node (E) and prompt an LLM to select additional nodes (as a

⁴Selecting background/belief nodes is performed recursively, where any node upstream of a selected node must also be selected.

User	Turn 1 (A,B)
My steaks keep turning out kind of mushy. Why am I not getting a good char on them?	
Assistant	Turn 2 (E,I)
The most common issue is that your pan or grill isn't hot enough. Another big reason is excess moisture on the surface of the steak, which can prevent a good char from forming.	
User	Turn 3
Could cooking too many steaks at once be part of the problem?	
Assistant	Turn 4
What kind of pan or cooking surface are you using?	
User	Turn 5 (C)
I'm using a cast iron pan.	
Assistant	Turn 6 (F)
Yes—overcrowding the pan is definitely another possible issue.	
User	Turn 7 (D,G)
I thought steaks were supposed to stay kind of wet so they don't dry out so I've been dampening them, why does extra moisture stop them from getting a good char?	
Assistant	Turn 8 (H)
Because if there's too much moisture on the surface, you're essentially steaming the meat instead of searing it, which keeps that good char from developing.	

Figure 3: A full dialog transcript created through our graph traversal method from the graph in Figure 2.

well-formed subtree) to include in the answer. Here, the LLM identifies node I as necessary. Finally, we prompt an LLM to rewrite the list of statements into a fluent answer.

Turn 3: Follow-Up Question #1. Next, we select an unvisited edge from the answer subtree from which to create a **follow-up question**. In our example, we randomly choose the Substitution edge from E to F. We prompt an LLM with the transcript so far, the new top-level answer (F), and the DDA label, which guides the type of question being written. The LLM generates a follow-up question which becomes **turn 3**. Then, we select the relevant answer subtree (F) as in turn 2.

Turns 4-6: Clarifying Question. The selected answer subtree has a node with an unvisited, upstream background node (C). We treat these in-

stances of missing background information as an opportunity to generate a **clarifying question** where the assistant requests missing background information that is pertinent to the QUD.⁵ We prompt an LLM with the current transcript and the missing background information and generate both the clarifying question from the assistant and the response from the user, forming **turns 4 and 5** respectively; and generate the answer as before for **turn 6**.

Turns 7-8: Follow-Up Question #2. Next, we repeat the process from turn 3 and select the other unvisited edge (Motivation) to create a follow-up question. Once again, we generate a follow-up question using the transcript and the top-level answer (H) and select the relevant answer subtree (just H). Here, the answer subtree has an upstream unvisited belief node (G). We treat this differently from background nodes because knowing the user's belief states does not necessarily help answer the question, so a clarifying question is not necessary. Instead, we include the belief node in the LLM call that writes the follow-up question, producing the question in **turn 7** which includes the user's belief. Finally, we generate the answer as before.

5 Evaluation Framework

A key contribution of this work is a new annotation scheme for measuring the quality of information-seeking dialogs between a user and an assistant. Previous work on dialog evaluation has produced a wide range of quality criteria, including automatic reference-based metrics, human judgments of fluency, coherence, and relevance, and task-oriented measures (Deriu et al., 2021; Yeh et al., 2021). However, these criteria fail to capture whether a conversation is efficient as a whole; that is, whether it satisfies the user's information need without unnecessary repetition, ambiguity, or wasted turns.

Our annotation scheme tries to address this gap by operationalizing Gricean conversational maxims into a taxonomy of concrete problem types at the turn level (for both the user and the assistant) and the claim level (for the assistant), complemented by holistic quality ratings at the dialog level. Each annotation criterion targets a specific violation of Gricean maxims, summarized in Table 2. At the turn and claim level, our taxonomy decomposes dialog quality into identifiable problem types, enabling fine-grained diagnosis of where and how

⁵This approach is very general, encompassing various forms of ambiguity that can motivate clarifying questions as described in Tang et al. (2025).

Level	Problem Type	Maxim	Description
User Turn	Repetitive Question	Quantity	Asks a question already answered earlier in the dialog.
	Contradictory Statement	Quality	Contradicts background information from a previous turn.
	Failure to Respond	Relation	Ignores a clarifying question posed by the assistant.
Asst. Turn	Incomplete Answer	Quantity	Fails to fully address the user’s question.
	Dense/Confusing Ans.	Manner	Response is too dense or uses confusing language.
	Unhelpful Clarifying Q.	Relation	Asks an unhelpful or irrelevant clarifying question.
Asst. Claim	Repetitive Claim	Quantity	Repeats information already provided earlier in the dialog.
	Irrelevant Claim	Relation	Not relevant to answering the user’s question.
	Unnecessary Claim	Quantity	Relevant but not necessary to answer the question.
	Problematic Claim	Quality/Manner	Claim is wrong, strange, or out of place.
Level	Quality Rating	Maxim	Description
Dialog	Asst. Helpfulness	Relation	Assistant’s responses are useful to the user.
	User Humanness	Manner	User’s turns sound natural and human-like.
	Asst. Comprehensiveness	Quantity	Assistant thoroughly covers the user’s information need.
	Conv. Efficiency	Quantity	Information need is met without unnecessary turns.
	Conv. Flow	Manner	Conversation progresses naturally and coherently.

Table 2: Summary of the annotation scheme. Turn- and claim-level criteria identify specific problem types (binary); dialog-level criteria are holistic quality ratings on a 5-point Likert scale.

conversations lose efficiency (Table 3). We complement this with dialog-level Likert ratings that capture overall impressions of helpfulness, comprehensiveness, efficiency, and flow (Table 4). Complete annotation guidelines are included in Appendix E.

6 Experiments

Experimental Setup. We generate 105 dialogs in the cooking domain following the method described in §4, and each dialog is independently annotated by two annotators using the scheme described in §5. Annotators are graduate students, interns, and faculty in computer science and linguistics, as well as industry researchers and practitioners.

For initial comparison, we also annotate a selection of dialogs from MTRAG (Katsis et al., 2025), a multi-turn conversational search dataset in which human annotators produce user turns and modify LLM-generated assistant turns. Unlike our cooking-related dialogs, MTRAG was created from a variety of domains including Wikipedia pages (Rosenthal et al., 2024), financial Stack Exchange posts (Maia et al., 2018), government domain webpages, and technical documentation pages of a major cloud provider. As the dataset was originally intended for evaluating retrieval-augmented generation, some conversations have turns marked as unanswerable, which we filter out. We also filter dialogs longer than seven turns to match the length of the graph-generated dialogs, leaving 16 dialogs for evaluation. MTRAG also includes fully syn-

thetic dialogs, from which we sample another 16 with the same criteria.

Results. We report turn-level problem rates (Table 3) and dialog-level quality ratings (Table 4) across all evaluated datasets, aggregated over individual annotations (two per dialog). Our dialogs are preferred along most dimensions measured at the dialog level, with statistically significant wins for Enjoyment, Efficiency, and Flow. Our dialogs also have a lower rate of dense/confusing assistant answers, but a higher rate of irrelevant and unnecessary claims compared to MTRAG-Synth. Significance is measured using the two-proportion z-test for turn- and claim-level criteria and the Mann-Whitney U test (Mann and Whitney, 1947) for dialog-level ratings. To account for the difference in average number of turns, we perform a dialog length ablation in Appendix G, finding these trends hold regardless of dialog length.

The Efficiency Discrepancy. There is a mismatch between efficiency at the dialog level, where our dialogs are preferred, and at the claim level (irrelevant/unnecessary), where MTRAG-Synth is preferred. We hypothesize this may stem from differences in query granularity: MTRAG dialogs tend toward broader, more exploratory questions where claims are less likely to be flagged as unnecessary, while our graph dialogs contain focused questions with a narrower range of relevant claims.

To investigate this further, two authors annotated goal shifts between consecutive turns for

Level	Problem Type	Problem Rate (%)			Inter-Annotator Agreement (AC1)		
		Ours	MTRAG Human	MTRAG Synth.	Per-Prob. Type	Problem Detection	Prob. Type Classif.
User Turn	Repetitive Question	11.2	5.21	9.00	0.875	0.849	0.943
	Contradictory Statement	1.24	0.89	0.63	0.977		0.943
	Failure to Respond	0.36	0	0	0.996		1
Asst. Turn	Incomplete Answer	11.7	13.47	3.42	0.860	0.683	0.716
	Dense/Confusing Answer	7.46	16.35	9.93	0.848		0.843
	Unhelpful Clarifying Question	0.55	0	0	0.993		0.909
Asst. Claim	Repetitive Claim	4.85	3.43	4.64	0.929	0.657	0.819
	Irrelevant Claim	3.89	7.33	0.43	0.922		0.461
	Unnecessary Claim	9.84	14.57	7.05	0.831		0.288
	Problematic Claim	1.85	4.28	0.19	0.967		0.783
	Turns	3.59	6.5	5.44			
	Claims Per Turn	4.87	7.28	5.80			
	Stat. sig. wins	1/10	0/10	4/10			

Table 3: Turn- and claim-level results: problem type rates from human annotation (left) and inter-annotator agreement (right). Best problem rate per row is underlined; statistically significant improvements are **bolded**. Agreement is measured per individual problem type, as binary problem detection (any problem vs. none), and as problem type classification (which problem, conditional on both annotators flagging one).

a subset of dialogs (10 of each type), labeling each as initial, no-shift, minor-shift, or major-shift, with high agreement. Results show a striking difference: graph dialogs heavily favor minor shifts (50.5%) or no change (6.8%), with only 11.6% major shifts. MTRAG-Synth dialogs, on the other hand, favor major shifts (49%), followed by minor (30%) or no (1.8%) shifts. (Full statistics in [Appendix F](#).)

In graph dialogs, users decompose a broader information need into related subquestions across turns. A claim surfaced in one turn may be relevant to another, but annotators evaluating turns in isolation may mark it as irrelevant or unnecessary, lowering claim-level efficiency. MTRAG dialogs treat turns as independent atomic queries, producing targeted responses with fewer unnecessary/irrelevant claims per turn but without the progression of a natural conversation. This also illuminates the preference for graph dialogs in enjoyment and flow. Nearly half of MTRAG transitions introduce a significant shift, producing disjointed conversations, while graph dialogs more closely mirror how users organically explore a topic. This discrepancy can be further addressed in future work by characterizing the user’s information need within the dialog on a gradient of exploratory vs focused, and then taking the granularity of this need into account when interpreting evaluation scores.

IAA. We report inter-annotator agreement (IAA) via Gwet AC1 scores for the turn- and claim-level

Criteria	Rating (5-pt Likert)			IAA
	Ours	MTRAG Human	MTRAG Synth.	AC2
Helpfulness (A)	<u>4.17</u>	3.91	4.06	0.736
Humanness (U)	3.98	<u>4.0</u>	3.25	0.603
Enjoyment (A)	3.82	3.03	3.13	0.577
Comprehensiveness (A)	3.96	3.56	<u>4.0</u>	0.641
Efficiency	4.02	3.09	3.09	0.582
Flow	3.92	3.13	2.94	0.599
Stat. sig. wins	3/5	0/5	0/5	

Table 4: Dialog-level ratings (5-point Likert) and inter-annotator agreement (Gwet’s AC2). Best score per row is underlined; statistically significant improvements are **bolded**. A = Assistant, U = User.

criteria in [Table 3](#)) and AC2 scores for the dialog-level criteria in [Table 4](#) (Gwet, 2014). We obtain substantial (or better) agreement across the board. For a deeper analysis of the turn- and claim-level agreement, we introduce two additional agreement measures. *Problem detection* measures whether annotators agree that a turn or claim exhibits any problem at all, collapsing all problem types into a binary judgment. *Problem classification* measures, among cases where both annotators flagged a problem, whether they agree on the type of problem. The results show that annotators generally agree on both the presence and the type of problem for user turns. However, problem classification agreement is weaker for assistant claims, particularly for irrelevant and unnecessary, suggesting some ambiguity in distinguishing between these two types. We be-

Level	Metric	Per-Prob. Type	Problem Detection	Prob. Type Classif.
User Turn	Repetitive Question	0.910		1.000
	Contradictory Statement	0.997	0.906	1.000
	Failure To Respond	1.000		1.000
Assistant Turn	Incomplete Answer	0.752		0.934
	Dense/Confusing Answer	0.995	0.744	0.968
	Unhelpful Clarifying Question	0.995		1.000
Assistant Claim	Repetitive Claim	0.829		0.924
	Irrelevant Claim	0.986		0.962
	Unnecessary Claim	0.326	0.039	0.762
	Problematic Claim	0.949		0.962

Table 5: Agreement between human annotators and LLM annotations on all binary metrics (GPT-5.2). Agreement is measured per individual problem type, as binary problem detection (any problem vs. none), and as problem type classification (which problem, conditional on both annotators flagging one). Gwet’s AC1 is used for all measures.

Metric	Gwet’s AC2
Helpfulness (A)	0.788
Humanness (U)	0.874
Enjoyment (A)	0.788
Comprehensiveness (A)	0.340
Efficiency	0.874
Flow	0.867

Table 6: Agreement between human annotators and LLM annotations (GPT-5.2) on Likert scale metrics.

lieve that this can be addressed through improved annotation guidelines in future work, and that refining our notions of irrelevant and unnecessary at the claim level may also address the observed efficiency discrepancy.

LLM Annotations. In addition to human annotation using our rubric, we also test an LLM’s ability (GPT-5.2) to annotate dialogs using our evaluation framework. Tables 5 and 6 contain the IAA scores between human annotators and GPT-5.2. Agreement is measured by choosing the best agreement among the pool of human annotators, so an annotation of a metric is considered agreed if at least one human annotator agrees with the LLM. While LLMs are in agreement on most of the annotation subtasks (0.75-0.99), they have poor agreement for metrics associated with the necessity of claims (0.32) and the comprehensiveness of answers (0.34), further highlighting the gap in the ability of current proprietary LLMs to accurately identify user need and provide efficient answers.

7 Conclusion

We propose an approach to generate natural conversational transcripts for information-seeking dialogs.

We transform forum posts into rich dialogs, using an LLM to apply structural transformations grounded in discourse theory. In a small-scale study, annotators tend to prefer dialogs simulated by our graph-based method to other comparable approaches. This finding supports the utility of grounding dialog simulation in the Gricean maxims, QUD framing, and common ground theory.

Our dialog graph representation is highly flexible and extensible, leaving many open directions for future work. We may enhance graphs to incorporate *multimodal questions and answers* using images sourced from the texts of Stack Exchange posts. We may also incorporate comment threads, adding explicit follow-up/clarifying questions and answers as branches in the graph. We may also explore enriching graphs with external information: e.g., adding web-search turns or merging multiple graphs from related posts.

Our graphs provide transparent representations of claim dependencies, enabling subject matter experts to edit dialog content. As graphs scale to more complex topics, this modifiability can help bridge expert knowledge and the general public.

Finally, our synthetic data may be used for LLM training in two straightforward ways. First, we could use a DPO-style pairwise objective between our dialogs and dialogs generated to be less efficient, and optimize an LLM to prefer the more efficient dialog (Rafailov et al., 2023). Second, we could use our annotated synthetic dataset to train an effective LLM reward model that reflects human preferences, and use that model to perform standard RLHF training (Ouyang et al., 2022).

Limitations

Due to the limited scope of our study, our dataset is restricted to the Cooking Stack Exchange subsite. We have run preliminary evaluations with other sites (e.g. Travel) and the method appears to work well, but we have yet to run large-scale annotation. Also, the small size of our dataset and large size of our group of annotators means the annotations and results are highly variable. Finally, there is some risk of self-preference bias in our LLM-as-judge procedure since the same model is used for generation and evaluation (Panickssery et al., 2024).

Though we strove to maximize comparability between our generated dialogs and the MTRAG dialogs, there remain clear confounding factors. MTRAG dialogs still had more turns and more total words on average, and the discrepancy in topics prohibits dialog-to-dialog comparisons. Furthermore, the MTRAG dialogs were generated by an earlier and smaller model than the LLM used for our generation procedure. Future work should strive to achieve the closest possible comparison, controlling for all of these factors.

Finally, we did not extensively evaluate the intermediate graph representations, which may contain errors. We focused on evaluating the final dialogs, but evaluating graph quality is worth exploring in future work. Additionally, while our system does not include a method for inducing diverse *user personas* (Bao et al., 2023), incorporating a standard approach to persona modeling to our framework would be straightforward.

Acknowledgments

Thanks to the three anonymous reviewers for their very useful and insightful feedback. This work could not have happened without the contributions of our entire team of annotators: heartfelt thanks to members of the BLAST and LECS labs at CU Boulder, to students in the Spring 2026 graduate seminar on Computational Models of Discourse and Dialogue, and to Zach Palmer. This work was partly supported by a gift from Amazon.

References

- Vaibhav Adlakha, Shehzaad Dhuliawala, Kaheer Suleman, Harm de Vries, and Siva Reddy. 2022. [Top-iOCQA: Open-domain conversational question answering with topic switching](#). *Transactions of the Association for Computational Linguistics*, 10:468–483.
- Kaveri Anuranjana, Srihitha Mallepally, Sriharshitha Mareddy, Amit Shukla, and Radhika Mamidi. 2024. [Survey on computational approaches to implicature](#). In *Proceedings of the 21st International Conference on Natural Language Processing (ICON)*, pages 224–229, AU-KBC Research Centre, Chennai, India. NLP Association of India (NLP AI).
- Jianzhu Bao, Rui Wang, Yasheng Wang, Aixin Sun, Yitong Li, Fei Mi, and Ruifeng Xu. 2023. [A synthetic data generation framework for grounded dialogues](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10866–10882, Toronto, Canada. Association for Computational Linguistics.
- Jon Cai, Brendan King, Peyton Cameron, Susan Windisch Brown, Miriam Eckert, Dananjay Srinivas, George Arthur Baker, V Kate Everson, Martha Palmer, James Martin, and Jeffrey Flanigan. 2025. [In search of the lost arch in dialogue: A dependency dialogue acts corpus for multi-party dialogues](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20135–20149, Vienna, Austria. Association for Computational Linguistics.
- Lynn Carlson, Daniel Marcu, and Mary Ellen Okurovsky. 2001. [Building a discourse-tagged corpus in the framework of Rhetorical Structure Theory](#). In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*.
- Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wentaoh Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. [QuAC: Question answering in context](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184, Brussels, Belgium. Association for Computational Linguistics.
- Zhuyun Dai, Arun Tejasvi Chaganty, Vincent Y Zhao, Aida Amini, Qazi Mamunur Rashid, Mike Green, and Kelvin Guu. 2022. [Dialog inpainting: Turning documents into dialogs](#). In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 4558–4586. PMLR.
- Jan Deriu, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echevoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak. 2021. [Survey on evaluation methods for dialogue systems](#). *Artif. Intell. Rev.*, 54(1):755–810.
- Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. [Wizard of wikipedia: Knowledge-powered conversational agents](#). In *International Conference on Learning Representations*.
- Nouha Dziri, Sivan Milton, Mo Yu, Osmar Zaiane, and Siva Reddy. 2022. [On the origin of hallucinations in conversational models: Is it the datasets or the models?](#) In *Proceedings of the 2022 Conference of the North American Chapter of the Association for*

- Computational Linguistics: Human Language Technologies*, pages 5271–5285, Seattle, United States. Association for Computational Linguistics.
- Abed Alhakim Freihat, Mohammed R. H. Qwaider, and Fausto Giunchiglia. 2018. Using grice maxims in ranking community question answers. In *Proceedings of the Tenth International Conference on Information, Process, and Knowledge Management (eKNOW 2018)*, pages 38–43, Rome, Italy. IARIA.
- Karthik Gopalakrishnan, Behnam Hedayatnia, Qinlang Chen, Anna Gottardi, Sanjeev Kwatra, Anushree Venkatesh, Raefer Gabriel, and Dilek Hakkani-Tür. 2019. [Topical-chat: Towards knowledge-grounded open-domain conversations](#).
- Herbert P Grice. 1975. Logic and conversation. In *Speech acts*, pages 41–58. Brill.
- Milan Gritta, Gerasimos Lampouras, and Ignacio Iacobacci. 2021. [Conversation graph: Data augmentation, training, and evaluation for non-deterministic dialogue management](#). *Transactions of the Association for Computational Linguistics*, 9:36–52.
- Kilem L Gwet. 2014. *Handbook of inter-rater reliability: The definitive guide to measuring the extent of agreement among raters*. Advanced Analytics, LLC.
- Sanda Harabagiu, Steven Maiorano, Alessandro Moschitti, and Cosmin Bejan. 2004. [Intentions, implicatures and processing of complex questions](#). In *Proceedings of the Workshop on Pragmatics of Question Answering at HLT-NAACL 2004*, pages 31–42, Boston, Massachusetts, USA. Association for Computational Linguistics.
- Sanda M. Harabagiu and Dan I. Moldovan. 1996. [Textnet: A text-based intelligent system](#). In *AAAI Fall Symposium on Knowledge Representation Systems Based on Natural Language*, AAAI Technical Report FS-96-04, pages 32–42, Menlo Park, California. AAAI Press.
- Prathyusha Jwalapuram. 2017. [Evaluating dialogs based on Grice’s maxims](#). In *Proceedings of the Student Research Workshop Associated with RANLP 2017*, pages 17–24, Varna. INCOMA Ltd.
- Marten HL Kaas and Ibrahim Habli. 2025. Assuring ai safety: fallible knowledge and the gricean maxims. *AI and Ethics*, 5(2):1467–1480.
- Yannis Katsis, Sara Rosenthal, Kshitij Fadnis, Chulaka Gunasekara, Young-Suk Lee, Lucian Popa, Vraj Shah, Huaiyu Zhu, Danish Contractor, and Marina Danilevsky. 2025. [MTRAG: A Multi-Turn Conversational Benchmark for Evaluating Retrieval-Augmented Generation Systems](#). *Preprint*, arXiv:2501.03468.
- Wei-Jen Ko, Yating Wu, Cutter Dalton, Dananjay Srinivas, Greg Durrett, and Junyi Jessy Li. 2023. [Discourse analysis via questions and answers: Parsing dependency structures of questions under discussion](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11181–11195, Toronto, Canada. Association for Computational Linguistics.
- Mojtaba Komeili, Kurt Shuster, and Jason Weston. 2022. [Internet-augmented dialogue generation](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8460–8478, Dublin, Ireland. Association for Computational Linguistics.
- Lea Krause and Piek T.J.M. Vossen. 2024. [The Gricean maxims in NLP - a survey](#). In *Proceedings of the 17th International Natural Language Generation Conference*, pages 470–485, Tokyo, Japan. Association for Computational Linguistics.
- Hengli Li, Song-Chun Zhu, and Zilong Zheng. 2023. [Diplomat: A dialogue dataset for situated pragmatic reasoning](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 46856–46884. Curran Associates, Inc.
- Linxiao Li, Can Xu, Wei Wu, YUFAN ZHAO, Xueliang Zhao, and Chongyang Tao. 2020. [Zero-resource knowledge-grounded dialogue generation](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 8475–8485. Curran Associates, Inc.
- Lizi Liao, Le Hong Long, Zheng Zhang, Minlie Huang, and Tat-Seng Chua. 2021. [Mmconv: An environment for multimodal conversational search across multiple domains](#). In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’21*, page 675–684, New York, NY, USA. Association for Computing Machinery.
- Yen Ting Lin, Alexandros Papangelis, Seokhwan Kim, and Dilek Hakkani-Tur. 2022. [Knowledge-grounded conversational data augmentation with generative conversational networks](#). In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 26–38, Edinburgh, UK. Association for Computational Linguistics.
- Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. [Evaluating very long-term conversational memory of LLM agents](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13851–13870, Bangkok, Thailand. Association for Computational Linguistics.
- Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. 2018. [Www’18 open challenge: Financial opinion mining and question answering](#). In *Companion Proceedings of the The Web Conference 2018, WWW ’18*, page 1941–1942, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

- Henry B Mann and Donald R Whitney. 1947. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60.
- William C. Mann and Sandra A. Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text & Talk*, 8:243 – 281.
- Erik Miehling, Manish Nagireddy, Prasanna Sattigeri, Elizabeth M. Daly, David Piorkowski, and John T. Richards. 2024. [Language models in dialogue: Conversational maxims for human-AI interactions](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14420–14437, Miami, Florida, USA. Association for Computational Linguistics.
- Fengran Mo, Bole Yi, Kelong Mao, Chen Qu, Kaiyu Huang, and Jian-Yun Nie. 2024. [Convsg: Session data generation for conversational search](#). *Preprint*, arXiv:2403.11335.
- Biswesh Mohapatra, Gaurav Pandey, Danish Contractor, and Sachindra Joshi. 2021. [Simulated chats for building dialog systems: Learning to generate conversations from instructions](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1190–1203, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.
- Siru Ouyang, Zhuosheng Zhang, and Hai Zhao. 2021. [Dialogue graph modeling for conversational machine reading](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3158–3169, Online. Association for Computational Linguistics.
- Laura Panfili, Steve Duman, Andrew Nave, Katherine Phelps Ridgeway, Nathan Eversole, and Ruhi Sarikaya. 2021. [Human-ai interactions through a gricean lens](#). *Proceedings of the Linguistic Society of America*, 6(1):288–302.
- Arjun Panickssery, Samuel R. Bowman, and Shi Feng. 2024. [LLM evaluators recognize and favor their own generations](#). In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Rashmi Prasad, Bonnie Webber, and Alan Lee. 2018. [Discourse annotation in the PDTB: The next generation](#). In *Proceedings of the 14th Joint ACL-ISO Workshop on Interoperable Semantic Annotation*, pages 87–97, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Mohammed R. H. Qwaider, Abed Alhakim Freihat, and Fausto Giunchiglia. 2017. [TrentoTeam at SemEval-2017 task 3: An application of Grice maxims in ranking community question answers](#). In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 271–274, Vancouver, Canada. Association for Computational Linguistics.
- 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](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741. Curran Associates, Inc.
- Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. [CoQA: A conversational question answering challenge](#). *Transactions of the Association for Computational Linguistics*, 7:249–266.
- Craige Roberts. 2012. [Information structure in discourse: Towards an integrated formal theory of pragmatics](#). *Semantics and Pragmatics*, 5.
- Sara Rosenthal, Avirup Sil, Radu Florian, and Salim Roukos. 2024. [Clapnq: Cohesive long-form answers from passages in natural questions for rag systems](#). *Preprint*, arXiv:2404.02103.
- Heydar Soudani, Roxana Petcu, Evangelos Kanoulas, and Faegheh Hasibi. 2024. [A survey on recent advances in conversational data generation](#). *Preprint*, arXiv:2405.13003.
- Settaluri Sravanthi, Meet Doshi, Pavan Tankala, Rudra Murthy, Raj Dabre, and Pushpak Bhattacharyya. 2024. [PUB: A pragmatics understanding benchmark for assessing LLMs’ pragmatics capabilities](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12075–12097, Bangkok, Thailand. Association for Computational Linguistics.
- Robert Stalnaker. 2002. [Common Ground](#). *Linguistics and Philosophy*, 25(5/6):701–721.
- Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. [Dialogue act modeling for automatic tagging and recognition of conversational speech](#). *Computational Linguistics*, 26(3):339–374.
- Anfu Tang, Laure Soulier, and Vincent Guigue. 2025. [Clarifying ambiguities: on the role of ambiguity types in prompting methods for clarification generation](#). In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’25*, page 20–30, New York, NY, USA. Association for Computing Machinery.
- Qingyang Wu, Song Feng, Derek Chen, Sachindra Joshi, Luis Lastras, and Zhou Yu. 2022. [DG2: Data augmentation through document grounded dialogue generation](#). In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 204–216, Edinburgh, UK. Association for Computational Linguistics.

Matthias Wölfel, Mehrnoush Barani Shirzad, Andreas Reich, and Katharina Anderer. 2024. [Knowledge-based and generative-ai-driven pedagogical conversational agents: A comparative study of grice’s cooperative principles and trust](#). *Big Data and Cognitive Computing*, 8(1).

Ziang Xiao, Michelle X. Zhou, Q. Vera Liao, Gloria Mark, Changyan Chi, Wenxi Chen, and Huahai Yang. 2020. [Tell me about yourself: Using an ai-powered chatbot to conduct conversational surveys with open-ended questions](#). *ACM Trans. Comput.-Hum. Interact.*, 27(3).

Archana Yadav, Harshvivek Kashid, Medchalimi Sruthi, B JayaPrakash, Chintalapalli Raja Kullayappa, Mandala Jagadeesh Reddy, and Pushpak Bhattacharyya. 2025. [From recall to creation: Generating follow-up questions using bloom’s taxonomy and Grice’s maxims](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pages 1322–1338, Vienna, Austria. Association for Computational Linguistics.

Yi-Ting Yeh, Maxine Eskenazi, and Shikib Mehri. 2021. [A comprehensive assessment of dialog evaluation metrics](#). In *The First Workshop on Evaluations and Assessments of Neural Conversation Systems*, pages 15–33, Online. Association for Computational Linguistics.

Chujie Zheng, Sahand Sabour, Jiaxin Wen, Zheng Zhang, and Minlie Huang. 2023a. [Augesc: Dialogue augmentation with large language models for emotional support conversation](#). *Preprint*, arXiv:2202.13047.

Chujie Zheng, Sahand Sabour, Jiaxin Wen, Zheng Zhang, and Minlie Huang. 2023b. [AugESC: Dialogue augmentation with large language models for emotional support conversation](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1552–1568, Toronto, Canada. Association for Computational Linguistics.

Zilong Zheng, Shuwen Qiu, Lifeng Fan, Yixin Zhu, and Song-Chun Zhu. 2021. [GRICE: A grammar-based dataset for recovering implicature and conversational reasoning](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2074–2085, Online. Association for Computational Linguistics.

A Rhetorical Relation Labels

Our graphs use a subset of the DDA labels (Cai et al., 2025). Unlike DDA, where relations can be in either direction with respect to the original text, our relations are directed according to informational precedence. Thus, the exact definition of many labels is changed somewhat. We provide our full list of labels and the definitions (which are also provided in various LLM prompts) below.

1. **Answer** - a single statement that best answers the parent question.
2. **After** - the child statement happens after the parent statement temporally, often used for a sequence of steps in some process.
3. **Before** - the child statement precedes the parent statement temporally, often used for a sequence of steps in some process.
4. **Motivation** - the child statement motivates, justifies, or describes a purpose for the parent statement.
5. **Result** - the child statement is a result of the parent statement (but not its motivation). Results should be neutral or negative consequences, whereas positive consequences are typically labeled Motivation.
6. **Condition** - the child statement is a conditional with both an antecedent ("if X") and a consequent ("then Y").
7. **Manner** - the child statement describes the manner in which the parent statement should or does occur. Often used to answer "how" questions or provide details about the method for doing something.
8. **Similarity** - the child statement describes a similar situation to the parent statement.
9. **Contrast** - the child statement describes a contrasting situation to the parent statement.
10. **Instantiation** - the child statement provides a specific example of a general claim made by the parent statement.
11. **Level_of_Detail** - the child statement provides more or less specific information regarding the situation or scenario described in the parent statement.
12. **Exception** - the child statement presents one or more instances in which the described situation in the parent statement does not hold.
13. **Substitution** - the child statement presents some alternative to the parent statement.
14. **Concession** - the child statement denies or contradicts some assertion made in the parent statement due to some new circumstance or reason.

15. **Excerpt** - the child statement provides a quoted excerpt from an external source that provides evidence for the parent statement

B Graph Construction

We provide the full prompts for answer decomposition (Figure 5), initial graph construction (Figure 6), and adding background nodes (Figure 7) below. In some cases, we have truncated long examples for brevity.

C Example Dialogs

An example question/answer pair from Stack Exchange is given in Figure 4. We also provide two examples of dialogs generated with our method and their intermediate dialog graph about tomato sauce (Figure 8, Figure 9) and onions (Figure 10, Figure 11).

Question

I dropped a glass jar filled with rice. I've picked up the macro-sized pieces of glass, but am wondering if there is any way on earth to make the rice safe to eat, given that I am sure it contains small fragments and shards of glass. I could, naturally, go through it grain-by-grain, but this is only a couple of bucks worth of rice, and I don't think it warrants the time. Does anyone have any suggestions?

Answer

I would throw it out. The rice doesn't cost as much as a new digestive system. Sure, it's a waste. And I'm sure you would look very carefully. But the risk of one glass splinter still in it is existent. Therefore, I wouldn't try it.

Figure 4: An example question/answer from the Cooking Stack Exchange.

D Graph Traversal

In order to prevent extremely long dialogs, we model the number of turns as a *geometric distribution* with a tunable stopping probability p (we use $p = 0.2$). After each complete exchange (where the assistant has provided an answer) following the first, we randomly stop the conversation with the probability p .

When selecting edges, we use a depth-first ordering where edges outgoing from more recently visited parts of the graph are selected prior to those from less recently visited nodes. This prevents the conversation from wildly jumping around topics, instead following a natural progression that explores the answer one part at a time.

Finally, we incorporate a simple pruning procedure when adding new edges to the queue to process, where we use an LLM call to filter out edges to nodes that are off-topic for the overall goal of the dialog so far.

We provide the key prompts for traversal below, including the prompt for generating the main question (Figure 12), answers (Figure 13), follow-up questions (Figure 15), and clarifying questions (Figure 16), as well as the prompt for selecting the relevant answers for a given question (Figure 14).

E Annotation Guidelines

We provide the guidelines given to our annotators in Figure 17.

In addition to the above guidelines, we also provide an example of each issue the annotators are looking for. These examples are displayed in the following figures:

- Repetitive question - Figure 18
- Contradiction - Figure 19
- Ignored CQ - Figure 20
- Dense/confusing - Figure 22
- Incomplete/no answer - Figure 21
- Unhelpful CQ - Figure 23
- Repetitive Claim - Figure 24
- Not useful (Irrelevant) Claim - Figure 25
- Useful but not necessary claim - Figure 26
- Problematic/incorrect claim - Figure 27

F Goal Shift Annotations

Results of our goal shift annotations are found in Table 7.

Metric	Graph	MTRAG-Synthetic
No Goal Shifts (%)	6.80	1.80
Minor Goal Shifts (%)	50.50	30.00
Major Goal Shifts (%)	11.60	49.00
Evaluated Dialogues	10	10
Evaluated Turns Per Dialogue	5.3	5.6

Table 7: Results for goal shift annotation. Annotators rate the shift of the user goal between consecutive turns (no shift, minor shift, major shift). Agreement between annotators is high (0.718 Gwet's AC2).

G Dialog Length Ablation

The dataset of dialogs that we produce using our graph based dialog generation method (Section 4) are much shorter on average than the MTRAG (Katsis et al., 2025) dialogs that we evaluate against. As shown in Table 8 our Graph dialogs have, on average, roughly half the number of turns as MTRAG-Human and a third of the claims. Although our metrics are based on the proportion of issues and not absolute numbers, it could be speculated that the difference in the length of the dialogs is responsible for at least some of the effects we see in our experimental results. To account for this, we subsample a selection of our longer Graph dialogs (Graph-Long) and perform the same evaluation comparison which can be seen in Table 8 and Table 9. We notice that the results are similar to our evaluation on the larger dataset. However, Graph-Long has a much higher proportion of repetitive user questions and significantly less assistant answers which are dense and/or incomplete.

Level	Problem Type	Problem Rate (%)			Graph
		Graph-Long	MTRAG Human	MTRAG Synth.	
User Turn	Repetitive Question	17.1	<u>5.21</u>	9.00	11.2
	Contradictory Statement	2.05	0.89	<u>0.63</u>	1.24
	Failure to Respond	<u>0</u>	<u>0</u>	<u>0</u>	0.36
Asst. Turn	Incomplete Answer	6.6	13.47	<u>3.42</u>	11.7
	Dense/Confusing Answer	5.6	16.35	9.93	7.46
	Unhelpful Clarifying Question	0.8	<u>0</u>	<u>0</u>	0.55
Asst. Claim	Repetitive Claim	4.75	<u>3.43</u>	4.64	4.85
	Irrelevant Claim	3.05	7.33	0.43	3.89
	Unnecessary Claim	10.8	14.57	7.05	9.84
	Problematic Claim	0.62	4.28	<u>0.19</u>	1.85
Average Turn Count		6.28	6.5	5.43	3.59
Average Claim Count		23.28	45	29.16	15.17
dialog Count		25	16	16	105
Stat. sig. wins		1/10	0/10	2/10	

Table 8: Turn- and claim-level results: problem type rates from human annotation. Best problem rate per row is underlined; statistically significant improvements are **bolded**.

Criteria	Rating (5-pt Likert)			Ours
	Ours	MTRAG Human	MTRAG Synth.	
Helpfulness (A)	<u>4.28</u>	3.91	4.06	<u>4.17</u>
Humanness (U)	<u>4.0</u>	<u>4.0</u>	3.25	3.98
Enjoyment (A)	3.9	3.03	3.13	3.82
Comprehensiveness (A)	4.08	3.56	<u>4.0</u>	3.96
Efficiency	<u>4.12</u>	3.09	3.09	4.02
Flow	3.94	3.13	2.94	3.92
Average Turn Count	6.28	6.5	5.43	3.59
Average Claim Count	23.28	45	29.16	15.17
dialog Count	25	16	16	105
Stat. sig. wins	3/5	0/5	0/5	

Table 9: Dialog-level ratings (5-point Likert). Best score per row is underlined; statistically significant improvements are **bolded**. A = Assistant, U = User.

You will be given the raw text of a question and answer from a forum site. You are to split and rephrase the answer text into a sequence of statements with the following constraints:

- The statements should cover all of the information from the original answer
- You should not invent any new statements that are not directly stated in the text, even if they are implied/presupposed by the raw answer
- If there are multiple similar statements, only output a single one. Feel free to combine multiple statements into a single one.
- If there is a direct quote from some source, do not split it up into multiple statements. Instead, create a single statement that includes the direct quote, and prefix the quoted part with Markdown quotes >.

Please output a list of statements, where each statement has two fields:

- The "text" of the statement. The text should be close to the original text, but may be rephrased so it can stand on its own.
- The "type" of statement. Statements should either be "claim", "anecdote", "meta", "belief", "question". A "claim" statement describes a concrete, verifiable, widely-known fact that addresses the topic under discussion. An "anecdote" describes a personal story from the answerer, that is concrete but not necessarily widely-known. A "meta" statement references the question or answer itself, the forum site, other questions on the site, and any other metatextual information that is not necessary to ask the main question. A "belief" statement describes beliefs or preferences held by the answerer, and often uses phrases like "I think that...", "I doubt that...", and "I like to..."; however, you should strip these phrases and write the belief statement using impersonal words. A "question" is a clarifying question raised to the original asker.

For example, given the following question and answer:

Question:

Is the Bell Labs place where Unix was created still around to visit? I'm thinking of a trip to the US.

Answer:

Please tag your post correctly.

The Murray-Hill site in New Jersey is still in existence (it's part of Nokia Bell Labs now), but I doubt the Unix lab is preserved. You can visit the Bell Labs Technology Showcase, but that's a museum on the site, not the original offices.

... (truncated for this paper)

You should output the following statements:

```
{
  "statements": [
    {
      "text": "The asker should tag the post correctly.",
      "type": "meta"
    },
    {
      "text": "The Murray Hill site in New Jersey (where Unix was created) still exists.",
      "type": "claim"
    },
    ... (truncated for this paper)
  ]
}
```

Please produce claims for the following question and answer:

Question: \$question

Answer: \$answer

Figure 5: Prompt for answer decomposition into statements. The question decomposition prompt is very similar.

You are creating a directed graph (specifically a tree) representing information flow in a post from a question answering forum. I will give you a list of nodes, which represent the statements/claims that make up the question and answer. You are to create labeled directed edges between nodes that represent the relations between statements. Specifically, a directed edge from node A to node B means that node A should always come before node B in a conversation because the information introduced by node A must be introduced to the common ground prior to the information in node B. Edges may use any of the following rhetorical relation labels:

Answer - a single statement that best answers the parent question.

... (truncated for paper, see above for full list)

You will be given a list of numbered nodes, with the first node representing the question, and the remaining nodes representing the statements that make up the answer. Your job is to identify the relationship between nodes, given the following constraints and guidelines.

1. The question node should have exactly one outgoing edge to a node from the answer. This node should be the most important statement in the answer.
2. Every other node will be a descendent of this main answer node. A directed edge from one statement to another represents giving additional information via one of the rhetorical relations listed above.
3. Directed edges should represent information precedence, where the parent must be provided before the child. These edges do *not* necessarily follow the linear order that the nodes are presented in.
3. Every answer node must have a single parent (incoming edge) but may have multiple children (outgoing edges). There is no ordering of the children of a node.

In addition to the claim nodes, you will also be given the original question and answer for reference.

Your output should consist of a list of edges in the following format:

i -- Relation --> j

where Relation is one of the possible rhetorical relationships, i is the label of the parent node, and j is the label of the child node.

Consider the following example:

Question:

Is the Bell Labs place where Unix was created still around to visit? I'm thinking of a trip to the US.

Plaintext Answer:

The Murray-Hill site in New Jersey is still in existence (it's part of Nokia Bell Labs now),
... (truncated for this paper)

Nodes:

0 - Is the Bell Labs place where Unix was created still around to visit? I'm thinking of a trip to the US.

... (truncated for this paper)

Output:

0 -- Answer --> 3

3 -- Level_of_Detail --> 1

... (truncated for this paper)

Now annotate the following question/answer:

Question:

\$question

Plaintext Answer:

\$answer

Nodes:

\$nodes

Figure 6: Prompt for initial graph construction.

You are helping to create an acyclic directed graph representing information flow in a post from a question answering forum. I will give you a list of nodes, which represent the statements that make up the question and answer. I will also give you a list of directed edges between these nodes that represents the relations between the question and statements in the answer.

Specifically, a directed edge from node A to node B means that node A should always come before node B in a conversation because the information introduced by node A must be introduced to the common ground prior to the information in node B. Edges may use any of the following rhetorical relation labels:

Answer - a single statement that best answers the parent question.

... (truncated for paper, see above for full list)

I will give you the raw question and answer, the list of nodes in the graph, and a list of edges in the following format:

i -- Relation --> j

where Relation is one of the possible rhetorical relationships, i is the label of the parent node, and j is the label of the child node.

Then, I will give you two lists of new nodes that have yet to be added to the graph. These nodes will represent statements made by the user asking the main question, which will serve two main purposes. The nodes in the Background list will represent concrete, verifiable information about the user's situation or the world that gives context to the question or answer. The nodes in the Belief list will represent statements about the user's beliefs or attitude towards the question under discussion. You are to return a new list of edges that connects these Background and Belief nodes to the existing graph and to each other. Like the existing edges, these edges represent information flow, where an edge from node A to node B means that the information in node A must be provided before node B in any conversational trajectory.

First, create edges using the rhetorical relations listed above that connect all of the Background and Belief nodes to one another. Use the following constraints:

1. Every Background or Belief node should have zero or one parent node.
2. Nodes may have any number of children. A child node should be dependent on the parent node, such that it would not make sense to state the child before the parent.

Next, connect the Background and Belief nodes to the existing graph by adding directed edges to existing question or answer nodes. The core goal is to identify existing nodes on the graph that require some background information to make sense. Use the following constraints:

1. A Background or Belief node may have any number of children nodes in the existing graph.
2. Every edge connecting a background node to a question or answer node should be labeled "Background" or "Belief" depending on the type of node.
3. An edge from a Background or Belief node to a question or answer node means that the former must be introduced to the conversation before the latter.
4. If a Background or Belief node is a parent of some question or answer node, all of the children of the question or answer node also depend on the upstream Background or Belief node. Thus, you do not need to add additional edges from the upstream Background or Belief node to each downstream answer node.
5. Likewise, if a Background or Belief is a parent of some question or answer node, all of the upstream Background or Belief nodes are also assumed to be necessary to the question or answer node.
6. Some Background or Belief nodes are not necessary for any claim. In this case, do not add any outgoing edges.
7. You should only use the "Background" and "Belief" labels for edges connecting a background/belief node to the existing graph. Edges between background/belief nodes should use the standard rhetorical relations listed above.

Consider the following example:

... (omitted for paper)

Now annotate the following question/answer:

Question: \$question

Plaintext Answer: \$answer

Nodes: \$nodes

Existing Graph: \$graph

Background: \$background_nodes

Belief: \$belief_nodes

Figure 7: Prompt for adding background nodes.

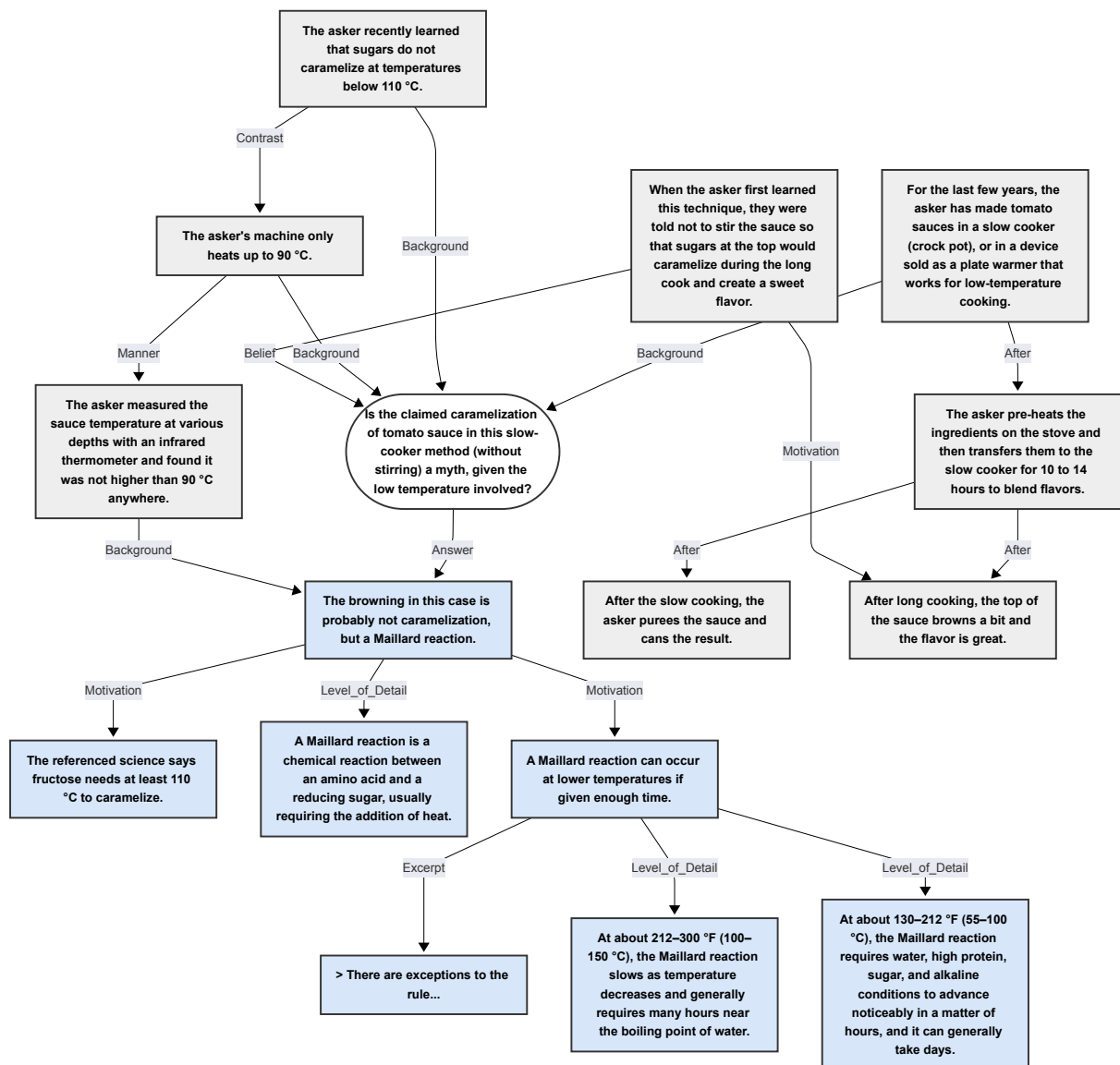


Figure 8: Dialog graph for a dialog about tomato sauce.

User	Turn 1
For the last few years I've been making tomato sauce in a slow cooker or a low-temperature "plate warmer" device that only heats to about 90 °C. When I first learned this method I was told not to stir so the sugars at the top would caramelize during the long cook and make the sauce sweeter, but I've since learned sugars don't caramelize below about 110 °C. Given the low temperature, is this supposed caramelization from not stirring basically a myth?	
Assistant	Turn 2
Have you actually checked what temperature the sauce reaches—especially at the surface versus deeper down—and what's the hottest reading you've seen?	
User	Turn 3
Yes—I measured at various depths/surfaces with an infrared thermometer, and nowhere in the sauce did it read higher than about 90 °C.	
Assistant	Turn 4
With a measured max around 90 °C, it's very unlikely you're getting true caramelization. Caramelization of common sugars generally requires temperatures above boiling, and fructose is often cited as starting around 110 °C under typical cooking conditions. If you're noticing darkening or "browned" flavor, a more plausible explanation is slow Maillard browning (reactions between sugars and amino acids), which can still occur in moist foods at lower temperatures given enough time—especially near the surface where the mixture can be a bit more concentrated.	
User	Turn 5
Is there a reliable reference that supports Maillard-type browning happening at temperatures around a simmer if it's cooked long enough?	
Assistant	Turn 6
From On Food and Cooking: > There are exceptions to the rule that browning reactions require temperatures above the boil. Alkaline conditions, concentrated solutions of carbohydrates and amino acids, and prolonged cooking times can all generate Maillard colors and aromas in moist foods. For example, alkaline egg whites, rich in protein, with a trace of glucose, but 90% water, will become tan-colored when simmered for 12 hours. The base liquid for brewing beer, a water extract of barley malt that contains reactive sugars and amino acids from the germinated grains, deepens in color and flavor with several hours of boiling. Watery meat or chicken stock will do the same as it's boiled down to make a concentrated demiglace. Persimmon pudding turns nearly black thanks to its combination of reactive glucose, alkaline baking soda, and hours of cooking; balsamic vinegar turns nearly black over the course of years!	

Figure 9: A full dialog transcript created through our graph traversal method from the graph in [Figure 8](#).

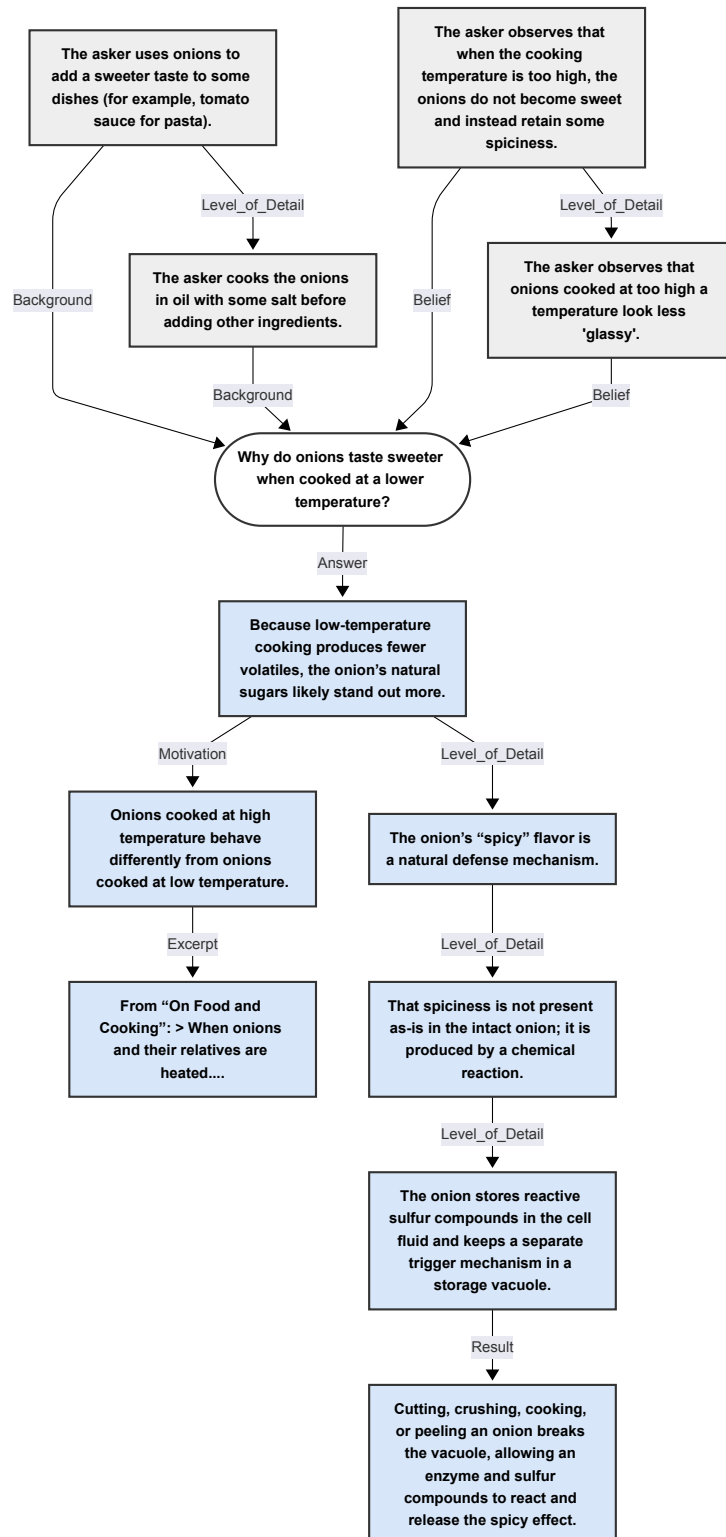


Figure 10: Dialog graph for a dialog about onions.

User	Turn 1
When I cook onions in oil with a bit of salt before adding other ingredients, I notice that if the heat is too high they don't turn as glassy, they keep some spiciness, and they don't get as sweet. I often use onions to add sweetness to dishes like tomato sauce for pasta. Why do onions taste sweeter when cooked at a lower temperature?	
Assistant	Turn 2
Onions cooked at high temperature behave differently from onions cooked at low temperature. At higher heat you drive off more pungent volatile compounds quickly and start pushing toward browning reactions, so they can stay sharper and not develop that mellow, sweet "glassy" character. With lower-temperature cooking, fewer volatiles are produced and lost, so the onion's natural sugars tend to stand out more, giving you a sweeter taste.	
User	Turn 3
Is there a reliable source that talks about how cooking temperature changes onion flavor?	
Assistant	Turn 4
From Harold McGee's <i>*On Food and Cooking*</i> : "When onions and their relatives are heated, the various sulfur compounds react with each other and with other substances to produce a range of characteristic flavor molecules. The cooking method, temperature and medium strongly affect the flavor balance. Baking, drying, and microwaving tend to generate trisulfides, the characteristic notes of overcooked cabbage. Cooking at high temperatures in fat produces more volatiles and a stronger flavor than do other techniques." (p. 311)	

Figure 11: A full dialog transcript created through our graph traversal method from the graph in [Figure 10](#).

You are simulating a user interacting with an AI chatbot. I will give you a list of background statements and the main question that the user is asking. You are to rewrite all of these into a few sentences that sound natural. For example, given: Background: - I have an Acura MDX - The Acura is from 2018 - The gas station is out of premium gas Question: Is it okay to use non-premium gas? You should respond with: I have a 2018 Acura MDX. If the gas station is out of premium gas, is it okay to use non-premium gas? Please respond only with the answer. The real background statements and question are below. Background: \$background Question: \$question	
---	--

Figure 12: Prompt for writing the initial question turn.

I will give you a dialog transcript between an AI assistant and a human user. I will give you a list of statements (distinct discourse contributions) that make up the answer, which you will rewrite to sound more natural in the context of the conversation. You should keep the content the same but may rewrite, combine sentences, and add transitional phrases. You should also omit any of the provided statements if they have already been stated in the dialog transcript.

For example, given the following:

User:

Does my Acura MDX always need premium gas?

Assistant:

<Missing>

Required statements:

- You may use non-premium gas if necessary
- Using non-premium gas can damage the engine over time

You should respond with:

Using non-premium gas can be damaging to your engine over time. However, if necessary, it's fine to use non-premium gas.

Please respond only with the answer. The real transcript and statements are below.

\$transcript

Assistant:

<Missing>

Required statements:

\$answer_statements

Figure 13: Prompt for writing answers.

You are constructing a simulated dialog between a user and an AI assistant where the trajectory of dialog turns is guided by the structure of an answer tree. I will give you the transcript of the dialog so far, which ends in a question asked by the user. In addition, I will give you an answer tree with a single top-level node, where each node represents a statement that makes a discrete contribution to the dialog (roughly a Common Ground Unit). Edges represent information precedence (or presupposition), where the parent node must be entered into the discourse before the child. Edges are labeled with the type of relation between the parent and child node. The possible relations are as follows:

Answer - a top-level answer, which must be included
... (truncated for paper, see above for full list)

I will give you the list of nodes in the graph in the format:
i - Text
where i is the index of the node and Text is the full statement.

I will also give a list of edges in the following format:
i -- Relation --> j
where Relation is one of the possible rhetorical relationships, i is the index of the parent node, and j is the index of the child node.

You are to identify the subgraph that is necessary to answer the question by providing a list of node indices. Consider Grice's conversational maxims: you should select all of the nodes which are necessary to answer the stated question comprehensively, but should not select any nodes which are not necessary. Furthermore, you can only select a node if all of its parents are also selected. You may consider the relation label, which often gives clues as to whether the edge should be included. The children of the root node are top-level answers and must be included. Output both the node indices and a short statement justifying why that node is necessary.

For example, given the following:

Transcript:

User:

How do I fill my car tires?

Nodes:

0 - Root

... (truncated for this paper)

Edges:

0 -- Answer --> 1

... (truncated for this paper)

You should output the following:

```
{
  "chosen_nodes": [
    {"idx": "1", "justification": "Top-level answer"},
    {"idx": "3", "justification": "The user wants to know specific methods to fill up their tires."}
  ]
}
```

Node 2 is not strictly necessary, as it gives extra details that the user might not care about.

Common reasons that a node would not be necessary include:

- The node provides a motivation/justification, but the question only asks a "what" or "how" question.
- The node provides historical context, but the user only cares about a practical, current topic.
- The node provides a specific example, but the question is general and can be answered without one.
- The node provides information that was already mentioned earlier in the dialog
- The node describes a related situation, but one that is distinct from the situation under discussion

Below are the real transcript and graph.

Transcript:

\$transcript

Graph:

\$graph

Figure 14: Prompt for selecting answer nodes.

I will give you a dialog transcript between an AI assistant and a human user. The dialog will end with a missing user turn that you are guessing. I will give you a short bullet-point summary of the AI's response, which will not include all of the details. Your goal is to write a general question in the context of the dialog that would elicit the AI response. The question should be a leading question that is clearly written with knowledge of the answer, but should be realistic based on the information so far in the dialog. The question should also be as simple and general as possible, and not multiple question clauses connected with commas. Questions should be neutral instead of leading questions that suggest the user already knows the answer.

To guide your generation, I will give you three additional pieces of information:

1. "Required Background" - a list of background statements that should be included in the user's turn in addition to the question that you write, which will describe the user's beliefs about the question.
2. "Target Statement" - The specific piece of information from the existing dialog that the user is asking a follow-up question about.
3. "Question Type" - A label for the type of question you should ask. The possible labels are as follows:

After - What happens after the target statement?

Before - What happens before the target statement?

... (truncated for paper)

For example, given the following:

(start example)

User:

How much creatine can I take safely?

Assistant:

Generally, it is recommended to take up to 5 grams of creatine daily. You can load more creatine for a short period of time, but should be careful since there are potential side effects.

User:

<Missing>

Assistant:

- You can load up to 20-25 grams per day safely
- You should only load for 5-7 days.

Required Background:

- The asker thinks they could take up to 100g of creatine

Target Statement: You can load more creatine for a short period of time.

Question Type: Manner

(end example)

You might predict the missing turn as:

Okay for loading can i take like 100g?

Please use the exact format I showed above, and only output the follow-up question. The follow-up question should be realistic, natural, and human-like, with possible typos or non-standard capitalization or punctuation. The real transcript is as follows:

\$transcript

User:

<Missing>

Assistant:

\$next_turn

Required Background:

\$beliefs

Target Statement: \$target_statement

Question Type: \$question_type

Figure 15: Prompt for writing follow-up questions.

You are simulating a user interacting with an AI chatbot. I will give you a transcript of the chat so far, where the last turn is a question asked by the user. The question requires some additional context to answer, which I will give you. You will write two turns: the assistant turn, which asks a clarifying question that requests the additional context, and the user's response, which provides that additional context. For example, given:

Transcript:

User:

Can I use non-premium gas in my car?

Additional Context:

- I have an Acura MDX

You should respond with:

Assistant:

What make and model is your car?

User:

It's an Acura MDX.

Please response with the format above, where the two turns are labeled "Assistant:" and "User:" with a newline before each message. The real transcript and additional context are below.

Transcript:

\$transcript

Additional Context:

\$background

Figure 16: Prompt for writing clarifying questions.

Project: Improving Conversational AI for Interactively Answering User Queries

Background. Many people use AI chatbots to answer specific questions, or to gather information about particular topics. Most of the time, chatbots respond to such questions with info dumps, generating pages of material. In this project, our goal is to generate examples of more efficient user-assistant dialogs that are also more natural and more interactive.

Your task. using these guidelines, evaluate generated dialogs, one conversational turn at a time. Note that we use “turn” in two ways. It can refer to one back-and-forth conversational interaction or to one participant’s side of that interaction. At the end, we ask a few questions about the entire dialog. Please ask any questions you might have, and remember there is no single right answer.

Turn-by-turn evaluation: different questions for User turn & Assistant turn

User Turn

1. **User goal:** in this turn, what is the user trying to understand, accomplish, learn, or learn about?
 - **Special for first user turn:** write down your understanding of the user’s overall goal in making this query, based on their first conversational turn.
 - **For the rest of the turns:** Is the user goal aligned with the goal for the previous conversational turn?
 - **Yes:** you feel the user is still aiming at the same goal as before
 - **No:** the user goal has changed significantly. In which case write down the new user goal and mark whether or not the previous goal was completely satisfied before this shift
2. **General issues:** By default, we assume that the user is being a good conversational partner. If, however, you notice any of the issues listed below in this conversational turn, click the associated checkbox. If you notice some other issue, please write it in the comment box.
 - **Repetitive question:** user asks a question already answered earlier in the dialog
 - **Contradiction:** user contradicts background information they previously provided
 - **Ignored CQ:** sometimes the assistant asks clarifying questions (CQ) to better understand the user’s goal. Select this box if the user has failed to answer a CQ from the previous asst. turn.

Assistant Turn

1. **General issues:** By default, we assume that the assistant has answered the user’s question in a reasonable way. If, however, you notice any of the issues listed below in this conversational turn, click the associated checkbox. If you notice some other issue, please write it in the comment box.
 - **Dense/confusing:** assistant provides information using overly dense or confusing language
 - **Incomplete/no answer:** assistant does not completely answer the user’s question
 - **Unhelpful CQ:** assistant asks clarifying question that is not helpful for answering the question
2. **Claim-by-claim issues:** the interface will show you a list of claims extracted from the assistant’s current conversational turn. For each claim, mark any issues you see with that claim. Again, you can use the comment box for issues not on this list.
 - **Repetition:** the information in this claim has already been provided earlier in the dialog
 - **Not useful:** this claim is not relevant or not at all useful for answering the current question
 - **Useful but not necessary:** this claim is useful or relevant to the topic, but not strictly necessary to answer the stated question
 - **Problematic/incorrect:** this claim is wrong, strange, or out of place

Whole dialog evaluation: statements about the conversation as a whole. This final stage of the annotation process asks you to respond to six statements about the conversation as a whole. For each statement, choose one of the following answers:

- **Strongly Agree:** statement holds throughout the dialog, for all turns
- **Agree:** statement holds for most turns in the dialog; he turns for which it doesn’t hold do not egregiously violate the statement
- **Neutral:** statement holds for about as many turns in the dialog as it doesn’t
- **Disagree:** either statement doesn’t hold for most of the turns in the dialog, or there are a few turns that egregiously violate the statement
- **Strongly Disagree:** statement doesn’t hold for any dialog turn

General notes

- When you finish annotating a turn, click Next Turn to move on to the next turn.
- Before submitting, you can use the Previous Turn button to go back to earlier annotations.
- Click Submit Annotations when you’ve finished the whole dialog evaluation
- Use the Comment Box for questions, observations, etc. - anything about the dialog you think is interesting or important and not covered by the structured evaluation
- If you have suggestions for improving the evaluation/annotation process, we’re interested in hearing those too! (but not in the comment box)

Figure 17: Annotation guidelines provided to annotators.

Example

User

Can I make banana bread with yeast instead of baking powder?

Assistant

Banana bread "can" be made with yeast, and it will rise-but baking powder is used instead mainly because it's faster and more convenient. Yeast takes longer to work (even instant yeast), so you have to wait for the dough to rise and usually do extra handling along the way. Yeast also changes the flavor. It can add a pleasant fermented complexity that many people enjoy, but it can also introduce a more distinctly "bread-like" or yeasty note that isn't always desirable in something people think of as cake-like. (Modern baker's yeast tends to produce less of that unwanted yeasty character than many historical yeasts did, but the effect can still be there.)

User

Does yeast change the flavor?

Assistant

Yeast can add a pleasant fermented complexity that many people enjoy, but it can also introduce a more distinctly "bread-like" or yeasty note that isn't always desirable in something people think of as cake-like. (Modern baker's yeast tends to produce less of that unwanted yeasty character than many historical yeasts did, but the effect can still be there.)

Explanation

The user question posed in the second turn should be marked as repetitive because the assistant asks a question about the effect of yeast with regard to flavor and this information was already provided by the assistant in the first turn.

Figure 18: Example of a repetitive question asked by the user.

Example

User

I have a lot of roses in my garden right now, and I'd like to make rose-flavored Turkish delight using homemade rosewater. I know eating flowers from the garden can be risky since some can be toxic or unpleasant if not processed correctly, and most rosewater recipes online seem geared more toward cosmetic fragrance than cooking. Besides washing the roses, what should I consider to make sure homemade rosewater is safe for cooking?

Assistant

One of the biggest safety concerns with any foraged ingredient is correct identification, but roses are generally straightforward to identify. The bigger practical risk is chemical exposure: roses that are cultivated or planted in hedges along farm fields may have been sprayed or exposed to pesticide drift. If your roses are home-grown and you haven't treated them with anything that isn't suitable for food crops, then a thorough rinse and a quick check for bugs is usually all you need.

User

When I go to the store to pick out the roses for making homemade rosewater, is there a specific type of rose I should buy?

Explanation

The user question posed in the second turn should be marked as contradicting because the user mentions they are going to the store to pick out roses, but in the first turn they mention using homemade roses.

Figure 19: Example of a contradictory question asked by the user.

Example

User
I'm trying to make big, fluffy Mexican- style flour tortillas at home. Mine keep coming out thin, crispy, and small. Any ideas ?
Assistant
What is the recipe you are using?
User
Would getting new equipment help?

Explanation

The user question posed in the second turn should be marked as ignoring the clarifying question because the assistant asks a clarifying question in the first turn that the user ignores when asking a question in the second turn.

Figure 20: Example of a user failing to respond to a clarifying question.

Example

User
Can I make banana bread with yeast instead of baking powder? I'm specifically wondering how it would affect the taste?
Assistant
Banana bread "can" be made with yeast, and it will rise-but baking powder is used instead mainly because it's faster and more convenient. Yeast takes longer to work (even instant yeast), so you have to wait for the dough to rise and usually do extra handling along the way.

Explanation

The question should be marked as incomplete, because part of the user question asks about how yeast would affect the taste of banana bread which the assistant does not provide an answer for.

Figure 21: Example of an incomplete assistant answer.

Example

User
I'm wondering if banana bread might rise better if it were made with yeast. If that's the case, why would you use baking powder instead of yeast-do people prefer it because they like banana bread denser or wetter?
Assistant
To interrogate the premise rigorously, it is useful to situate banana bread within the broader taxonomy of leavened baked goods, specifically the category commonly labeled quick breads. These occupy an intermediate conceptual space between chemically leavened cakes and biologically leavened yeast breads, and the distinction hinges not merely on the source of carbon dioxide but on the entire cascade of biochemical and rheological processes that accompany each leavening strategy. In principle, banana bread could indeed be leavened with yeast. Yeast (most commonly <i>Saccharomyces cerevisiae</i>) metabolizes fermentable sugars, producing carbon dioxide and ethanol, which in turn expand the gluten network and generate lift in the dough matrix. However, the operative word here is dough. Yeast - based systems implicitly assume the development of a viscoelastic gluten structure capable of trapping fermentation gases over time. Traditional banana bread batter, by contrast, is deliberately formulated to minimize gluten development through a combination of relatively high sugar content, significant fat, and a mixing regime designed to avoid elastic structure formation. These factors collectively push the product toward the tender, friable crumb associated with cakes rather than the resilient chew characteristic of yeast breads.

Explanation

The assistant response should be marked as too dense as the response unnecessarily complicates the answer.

Figure 22: Example of a dense/confusing assistant answer.

Example

User Can I make banana bread with yeast instead of baking powder?
Assistant What is the recipe you are using?

Explanation

This assistant clarifying question should be marked as unhelpful, because the actual banana bread recipe being used doesn't have a bearing on whether or not yeast can be used instead of baking powder and is distracting from the original user question.

Figure 23: Example of an unhelpful clarifying question asked by the assistant.

User Can I make banana bread with yeast instead of baking powder?
Assistant Banana bread "can" be made with yeast, and it will rise-but baking powder is used instead mainly because it's faster and more convenient . Yeast takes longer to work (even instant yeast), so you have to wait for the dough to rise and usually do extra handling along the way.
User Does yeast change the flavor?
Assistant Yeast can add a pleasant fermented complexity that many people enjoy, but it can also introduce a more distinctly "bread-like" or yeasty note that isn't always desirable in something people think of as cake-like. (Modern baker's yeast tends to produce less of that unwanted yeasty character than many historical yeasts did, but the effect can still be there). However, the main disadvantage of using yeast is that it lengthens the baking process significantly.
Claim <i>the main disadvantage of using yeast is that it lengthens the baking process significantly</i>

Explanation

The claim given above should be marked as a repetition because the assistant already provides the same information in the first turn.

Figure 24: Example of a repetitive claim in an assistant answer.

Example

User Can I make banana bread with yeast instead of baking powder?
Assistant Banana bread "can" be made with yeast, and it will rise-but baking powder is used instead mainly because it's faster and more convenient. Yeast takes longer to work (even instant yeast), so you have to wait for the dough to rise and usually do extra handling along the way. There are whole categories of yeasted cake recipes that were once common in Europe.
Claim <i>There are whole categories of yeasted cake recipes that were once common in Europe</i>

Explanation

The claim given above should be marked as not useful because the comment that yeasted cake recipes used to be common in Europe is not at all useful towards answering the user's question about substituting yeast for baking powder in banana bread recipes.

Figure 25: Example of a not useful (irrelevant) claim in an assistant answer.

Example

User

Does the sugar in banana bread affect how yeast behaves?

Assistant

Yes-sugar has a big effect on yeast. In general, high sugar levels in doughs and batters inhibit yeast activity, because the yeast has a harder time getting water and functioning normally. As a result, high-sugar cake-style batters tend to rise much more slowly, and some yeasted "cake" recipes are designed for a very long, cool fermentation-sometimes an overnight rise-so the yeast has time to develop and multiply. If you want to avoid this lengthy baking process, baking powder can be used instead of yeast to shorten the process significantly.

Claim

baking powder can be used instead of yeast to shorten the time needed for the baking process

Explanation

The claim given above should be marked as not useful but not necessary because a comparison of baking powder to yeast for banana bread may be relevant to the topic but is not necessary to answer the specific question.

Figure 26: Example of an unnecessary claim in an assistant answer.

Example

User

Does the sugar in banana bread affect how yeast behaves?

Assistant

Banana bread can be made with yeast, and it will rise-but baking powder is used instead mainly because it's faster and more convenient. Yeast takes longer to work (even instant yeast), so you have to wait for the dough to rise and usually do extra handling along the way. Some bakers also claim that bananas contain trace magnetic fields that can slightly interfere with yeast fermentation, though this idea isn't supported by mainstream food science.

Claim

Some bakers also claim that bananas contain trace magnetic fields that can slightly interfere with yeast fermentation, though this idea isn't supported by mainstream food science.

Explanation

The claim given above should be marked as problematic because it contains a strange, unfounded claim regarding magnetic fields impacting yeast fermentation.

Figure 27: Example of a problematic claim in an assistant answer..