Structured Chain-of-Thought Prompting for Few-Shot Generation of Content-Grounded QA Conversations

Md Arafat Sultan **Jatin Ganhotra** **Ramón Fernandez Astudillo**
IBM Research AI
{arafat.sultan,ramon.astudillo}@ibm.com
jatinganhotra@us.ibm.com

Abstract

We introduce a structured chain-of-thought (SCoT) prompting approach to generating content-grounded multi-turn question-answer conversations with a pre-trained large language model (LLM). At the core of our proposal is a structured breakdown of the complex task into a number of states in a state machine, so that actions corresponding to various subtasks, *e.g.*, content reading and utterance generation, can be executed in their own dedicated states. Each state leverages a unique set of resources, including prompts and (optionally) additional tools, to augment the generation process. Automatic evaluation shows that SCoT prompting with designated states for hallucination mitigation can increase agent faithfulness to grounding documents by up to 16.8%. When used as training data, our open-domain conversations synthesized from only 6 Wikipedia-based seed demonstrations train strong conversational QA agents. In out-of-domain evaluation, for example, we observe improvements of up to 13.9% in F_1-score against ground truth over target domain gold data when the latter is augmented with our generated examples.

1 Introduction

Despite enormous advances in large language model (LLM) capabilities in recent years, their notorious propensity to hallucinate, *i.e.*, generate text that are factually inconsistent with existing knowledge, remains an open issue (Xu et al., 2023b; Bang et al., 2023; Huang et al., 2023; Xu et al., 2024). Of particular interest to us in this paper are what OpenAI (2023) characterize as *closed-domain* hallucinations, whereby models fail to generate text that can be supported by a given document even when they are explicitly instructed to do so (Xu et al., 2023b; Qiu et al., 2023; Maynez et al., 2020). Despite being powerful generators, LLMs can thus be unreliable readers, especially in the absence of extensive instruction tuning.

Document:

Title: Harry Potter

Text: Harry Potter is a series of seven fantasy novels written by British author J. K. Rowling. The novels chronicle the lives of a young wizard, Harry Potter, and his friends Hermione Granger and Ron Weasley, all of whom are students at Hogwarts School of Witchcraft and Wizardry. The main story arc concerns Harry's conflict with Lord Voldemort, a dark wizard who intends to become immortal, overthrow the wizard governing body known as the Ministry of Magic and subjugate all wizards and Muggles (non-magical people). *Continued...*

Conversation:

User: Tell me about Harry Potter.

Agent: Harry Potter is a series of seven fantasy novels written by British author J. K. Rowling. It tells the story of Harry Potter and his friends, who are all young wizards.

User: What happens in the story?

Agent: The main story arc concerns Harry's conflict with Lord Voldemort. Voldemort is a dark wizard who intends to become immortal and subjugate all wizards and Muggles (non-magical people).

User: Does Harry have any siblings?

Agent: Sorry, I can't find an answer in the document.

User: Tell me more about his friends.

Agent: Hermione Granger and Ron Weasley are Harry's friends. They are all students at Hogwarts School of Witchcraft and Wizardry.

User: Are there any movies based on the series?

Agent: Sorry, I can't find an answer in the document.

Figure 1: A multi-turn QA conversation grounded in a document. If the document does not have an answer to a user query, the agent acknowledges so in its response.

Here we focus on closed-domain hallucination in pre-trained LLMs in the specific context of conversational question answering (QA) (Choi et al., 2018; Reddy et al., 2019; Adlakha et al., 2022), and its mitigation through inference-time augmentation. Concretely, given a document, we want to generate a multi-turn QA conversation between a *user* and an *agent*, in which the agent's responses to the user's queries must be grounded in the document. The requirements also crucially include for the agent to be able to determine if the given document has an answer to a query, and refrain from giving a memorized or made-up answer if not. Figure 1 shows an example. Compared to a single-turn setting (Sadat

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 16172–16187
November 12-16, 2024 ©2024 Association for Computational Linguistics

et al., 2023), multi-turn QA provides a more useful backdrop for studying LLM hallucinations, as an increased number of diverse user queries on a shared topic is likely to induce agent hallucinations more often than a single QA turn.

We propose novel algorithms for this task (§3) that adhere to the general notion of language model (LM) augmentation (Yao et al., 2023; Xu et al., 2023a; Schick et al., 2023), whereby a set of actions – each leveraging its own dedicated tools and resources – are executed in an interleaved manner to solve a complex problem. For example, besides generating utterances, we execute intermediate tasks that aim to answer questions such as: "Is this user query answerable from the given document?" or "Where in the document is the answer?" All subtasks are performed using few-shot in-context learning (ICL) (Brown et al., 2020), for which we utilize task-specific instructions, exemplars and (optionally) supporting models.

Our proposed actions and their execution sequences can be collectively represented as a state machine (§2), whose state transitions define our different algorithms (§3). We refer to this approach as structured chain-of-thought (SCoT) prompting, which, like ordinary chain-of-thought (CoT) prompting (Wei et al., 2022), computes a final output through a set of careful reasoning steps, but unlike CoT, distributes those steps across designated states of a state machine. One key advantage of the approach stems from the simplicity of its individual actions, which enables us to successfully prompt relatively small and open-source LLMs to generate high-quality conversations.

Leveraging open-source models such as FALCON-40B (Almazrouei et al., 2023) and FLAN-UL2-20B (Tay et al., 2023) and a small set of Wikipedia-based exemplars that we create by hand (§3) – the example of Figure 1 is one of them – we generate open-domain QA conversations from Wikipedia passages using our algorithms, and evaluate them both intrinsically and extrinsically (§4). In intrinsic evaluation, we directly examine the quality of the generated conversations, including their faithfulness to the grounding document and their overall accuracy relative to pseudo-references provided by a high-performance instruction-tuned MIXTRAL-8X7B-INSTRUCT-V0.1 (Jiang et al., 2024). Automatic evaluation using lexical and semantic overlap metrics shows that our proposed mechanisms for mitigating agent hallucination do indeed reduce it by up to 16.8%, improving overall accuracy of agent utterances by as much as 7.7%.

In extrinsic evaluation, we train agents with our generated data to answer questions in multi-turn QA conversations, and evaluate them against gold labels using automatic lexical overlap metrics. We report experiments with few-shot ICL and supervised fine-tuning (SFT) of agents on two conversational QA datasets: DoQA (Campos et al., 2020) and QuAC (Choi et al., 2018). In ICL evaluation, our synthetic data – generated from only 6 seed demonstrations and with a relatively small LLM – outperforms human-labeled data, including target domain gold data. We also observe strong performance in SFT evaluation, which includes training conversational QA agents only with synthetic data as well as augmenting existing target domain gold data with it. For example, augmenting with our open-domain synthetic data improves agent performance over using only target domain gold data by an absolute 10–14%.

In summary, the following are our main contributions: ***(i)*** We present novel *structured* chain-of-thought prompting methods with LM augmentation for generating document-grounded QA conversations using pre-trained LLMs; ***(ii)*** In automatic intrinsic evaluation, our proposed augmentations for hallucination mitigation help the LLM agent remain considerably more faithful to the given document; ***(iii)*** In automatic extrinsic evaluation on grounded conversational QA datasets, our generated conversations demonstrate strong standalone performance as well as the ability to effectively augment target domain gold data.

2 Preliminaries

We introduce our state machine for SCoT prompting in this section, and discuss the alignment of its states to different stages of generating a document-grounded multi-turn QA conversation. Further implementation details are provided in §3.

Let $\mathbb{D}$ be the set of all documents and $\mathbb{C}_N$ the set of QA conversations $\langle q_i, r_i \rangle_{i=1}^{N}$ of length N, where each q_i is a user utterance (a query) and r_i is a corresponding agent utterance (a response). Our task is to map an input document $d \in \mathbb{D}$ to a conversation $c_N = \langle q_1, r_1, ..., q_N, r_N \rangle \in \mathbb{C}_N$ such that c_N is grounded in d: $c_N \sim p_{C_N|D}(\cdot \mid d)$.

Our proposed algorithms prompt specific sets of state transitions in the state machine of Figure 2 to generate a single utterance pair (q_i, r_i); a full conversation c_N is generated by repeating the process

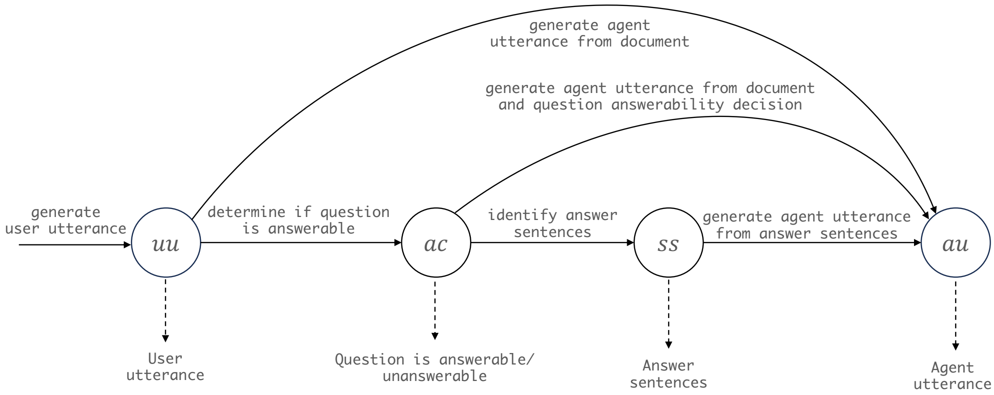

Figure 2: State machine for generating a single user-agent utterance pair within a multi-turn conversation (§2). An action (incoming arrow label) is executed in every state by few-shot prompting an LLM (§3), and an output is generated (dotted arrows). One of multiple possible transitions then takes place (solid arrows), depending on the algorithm being run. A grounding document and a conversation history (not in the diagram) are present in all steps.

N times. There are four states in the state machine, each corresponding to an individual action:

- **User utterance generation** ($\boldsymbol{uu}$): The next user utterance q_i in an ongoing conversation c_N is generated: $q_i \sim p_{uu}(\cdot \mid \langle q_j, r_j \rangle_{j=1}^{i-1}, d)$.
- **Question answerability classification** ($\boldsymbol{ac}$): The current user query q_i is classified as *answerable* or *unanswerable* from d: $a_i \sim P_{ac}(\cdot \mid q_i, \langle q_j, r_j \rangle_{j=1}^{i-1}, d)$. This state captures the notion that the assessment of the answerability of a user query can be kept separate from generating a response for it, which provides modularization and added flexibility. For example, a different set of resources can now be leveraged for question answerability classification, such as a classifier trained on existing QA data.
- **Answer sentence selection** ($\boldsymbol{ss}$): Information pertaining to a single user query q_i is often contained within a subset of all sentences in d. Before generating an agent response, it may be advantageous to identify those sentences so that response generation can focus more or solely on them (Sun et al., 2023a; Adolphs et al., 2022). In state $\boldsymbol{ss}$, the current query q_i is mapped to relevant sentences in d: $s_1, ..., s_M \sim P_{ss}(\cdot \mid q_i, \langle q_j, r_j \rangle_{j=1}^{i-1}, d)$.
- **Agent utterance generation** ($\boldsymbol{au}$): A response to the current query q_i is generated: $r_i \sim p_{au}(\cdot \mid q_i, \langle q_j, r_j \rangle_{j=1}^{i-1}, d^*)$. State $\boldsymbol{au}$ can be reached from any of the other three states; the following are to be noted: (*i*) If $\boldsymbol{au}$ is reached via a $\boldsymbol{uu} \rightarrow \boldsymbol{au}$ transition, then the input document $d^* = d$; (*ii*) If $\boldsymbol{au}$ is reached via an $\boldsymbol{ss} \rightarrow \boldsymbol{au}$ transition, then d^* may contain only the sentences of d deemed relevant to q_i in $\boldsymbol{ss}$, or alternatively special symbols marking those sentences; and finally, (*iii*) If $\boldsymbol{au}$ is reached via an $\boldsymbol{ac} \rightarrow \boldsymbol{au}$ transition, then this step is deterministic whenever q_i is deemed unanswerable in $\boldsymbol{ac}$, in which case r_i takes the form of a pre-defined *no answer* text.

Our methods can each be completely specified using (*i*) the state transitions it executes in the state machine, and (*ii*) the resources, *e.g.*, models and associated prompts, that it utilizes in its different states, as we detail in §3.

3 Methods

We implement five algorithms that execute three unique sequences of state transitions. What follows is a description of the sequences along with how the different algorithms implement them.

- $\boldsymbol{uu} \rightarrow \boldsymbol{au}$: One of our algorithms executes this transition to simply generate a user utterance first and then a corresponding agent utterance, both using a pre-trained LLM. The LLM is prompted with state-specific prefixes and manually created exemplars. Figure 3 shows an example 1-shot prompt for state $\boldsymbol{au}$ on the left; we use 2 exemplars in both states in practice. This prompt poses the task simply as one of text completion, which is what a pre-trained LLM is trained for. The prompt for $\boldsymbol{uu}$ works similarly, as shown in Figure 4 of Appendix A.
- $\boldsymbol{uu} \rightarrow \boldsymbol{ac} \rightarrow \boldsymbol{au}$: We implement two algorithms that execute this sequence. Both generate an actual agent response in state $\boldsymbol{au}$ only if the user's question is deemed answerable in state $\boldsymbol{ac}$, otherwise the response is a fixed pre-defined string that indicates no answer. Both algorithms use a pre-trained LLM in states $\boldsymbol{uu}$ and $\boldsymbol{au}$. However,

```
Here is a document broken down into its title and text:
Title: Harry Potter
Text: Harry Potter is a series of …
EoD
What follows is a conversation between a User and an
Agent. The User asks a number of different questions
about the topic of the document. The Agent answers those
based solely on the content of the document. If the
document does not contain information that can answer the
User's query, then the Agent says "Sorry, I can't find an
answer in the document." Here is the conversation:
User says: Tell me about Harry Potter. EoU
Agent says: Harry Potter is a … EoA
…
User says: Are there any movies based on the series? EoU
Agent says: Sorry, I can't find an answer in the
document. EoA

Here is a document broken down into its title and text:
Title: …
Text: …
EoD
What follows is a conversation between a User and an
Agent. The User asks a number of different questions
about the topic of the document. The Agent answers those
based solely on the content of the document. If the
document does not contain information that can answer the
User's query, then the Agent says "Sorry, I can't find an
answer in the document." Here is the conversation:
User says: … EoU
Agent says: … EoA
…
User says: … EoU
Agent says:
```

```
You are given a document with its title and
individual sentences. Each sentence is marked
with a sentence identifier (starting with S1).
You are also given an ongoing conversation
between a User and an Agent. Find the sentences
in the document using their identifiers, if any,
that can help answer the last user query in the
conversation.

Document:
Title: Harry Potter
S1: Harry Potter is a series of …
S2: The novels chronicle the life of …
S3: The main story arc concerns …
…
EoD
Conversation:
User says: Tell me about Harry Potter
Agent says: Harry Potter is a …
User says: What happens in the story?
Are there sentence(s) in the above document that
can help answer this last User query? Yes: S3

Document:
Title: …
S1: …
…
EoD
Conversation:
User says: …
Agent says: …
…
User says: …
Are there sentence(s) in the above document that
can help answer this last User query?
```

Figure 3: Prompts for states au and ss. **Left:** Agent utterance generation (au) with a pre-trained LLM. **Right:** Answer sentence selection (ss) with an instruction-following LLM. This diagram only shows 1-shot prompts for brevity; we use more demonstrations in practice (see Appendix A).

the two differ in their implementation of ac, for which one utilizes the same LLM and the other leverages a separate classifier fine-tuned on pre-existing QA data. Separate prompts containing both answerable and unanswerable exemplars are used for the two types of models, which we show in Figures 5 and 6 of Appendix A.

- $uu \to ac \to ss \to au$: Finally, two of our algorithms execute all four steps of the state machine. Upon classifying a question as answerable in state ac, relevant sentences are selected from the grounding document in state ss, which is provided as input to the step of au. One of the algorithms utilizes the same pre-trained LLM in all four states; the other uses an instruction-tuned model in states ac and ss as these two states correspond to actions that are more akin to classification tasks. Figure 3 shows a prompt for ss that instructs a FLAN-UL2-20B model to select sentences using their identifiers.

The above methodology enables us to study the reading and generation aspects of our task separately and in a controlled manner, and better understand where pre-trained LLMs need the most assistance. For example, we can explore the following questions directly: (*a*) Is it useful to break down the task into reading (ac, ss) and generation (uu, au) stages where SCoT prompting can be utilized? and (*b*) In the reading and reasoning states of ac and ss, can a pre-trained LLM perform the corresponding tasks by itself, or is a different set of tools needed?

4 Experiments

We write 6 simple Wikipedia-based QA conversations with a total of 32 user turns and an equal number of agent turns; 20 of the user queries are answerable and 12 are unanswerable from their respective passages. The example of Figure 1 is representative of the distribution; other titles include "Table Tennis" and "Evolution". We then prompt LLMs with these as exemplars to synthesize new data from additional Wikipedia passages.

In this section, we first analyze and evaluate our generated datasets intrinsically. To assess their practical utility, we then train conversational QA agents with each and evaluate those agents on different test sets. As stated before, each generation algorithm is uniquely specified by its three components: (*a*) the ***generator:*** we use FALCON-40B to generate all user and agent utterances; (*b*) the ***state transition sequence:*** the sequence of state transitions taken by the algorithm in the state machine; and (*c*) the ***assistant:*** some algorithms addition-

State Transitions	Assistant	% Has Answer	% Extracted	Faithfulness Lexical	Faithfulness WeCheck	Cls-Acc-MIXTRAL-I A	Cls-Acc-MIXTRAL-I UA	Cls-Acc-MIXTRAL-I HM	F1-MIXTRAL-I A	F1-MIXTRAL-I UA	F1-MIXTRAL-I HM
$uu \to au$	N/A	74.9	19.0	83.3	71.8	87.8	44.0	58.6	46.3	44.0	45.1
$uu \to ac \to au$	None	69.6	17.3	83.8	72.8	82.4	49.4	61.8	43.0	49.4	46.0
	FLAN-UL2-20B	47.6	20.9	93.6	86.5	68.7	85.9	76.3	37.8	85.9	52.7
$uu \to ac \to ss \to au$	None	58.4	58.5	90.5	80.6	67.1	55.5	60.7	30.7	55.5	39.5
	FLAN-UL2-20B	51.9	50.8	96.5	88.6	74.2	85.6	79.5	38.2	85.6	52.8

Table 1: Key statistics and LLM-as-reference evaluation results for data synthesized by a few-shot prompted FALCON-40B model, optionally assisted by a few-shot prompted FLAN-UL2-20B model in states $\boldsymbol{ac}$ and $\boldsymbol{ss}$ (if applicable). Accuracy of classification into answerable/unanswerable classes (Cls-Acc) and F1-scores are computed against pseudo-references provided by a prompted MIXTRAL-8X7B-INSTRUCT-V0.1 model. A/UA: the class of answerable/unanswerable questions; HM: the harmonic mean of the two previous columns.

ally use a FLAN-UL2-20B model in states $\boldsymbol{ac}$ and $\boldsymbol{ss}$ (if applicable) to augment generation. We use nucleus sampling (p=.9) (Holtzman et al., 2020) for user utterance generation and greedy decoding in all other steps. Details of model prompting at different stages of generation are provided in Appendix A.

All evaluations reported in this section are automatic, where we compare generated texts algorithmically with references such as human-written utterances or grounding documents. While human assessment of the generated texts is almost always more accurate at the level of individual texts, we opt for large-scale and extensive (both intrinsic and extrinsic) automatic evaluation instead, as long multi-turn conversations are expensive to annotate at scale for the results to be statistically reliable. Moreover, the grounded nature of our task inherently limits the extent to which an accurate response can differ from its ground truth.

4.1 Intrinsic Evaluation and Analysis

We sample $1,000$ Wikipedia passages and generate a conversation from every passage using each of our five algorithms; every conversation consists of 5 user and 5 agent utterances. Table 1 shows the results of our intrinsic analysis of this data. The first two columns provide two basic statistics: the % of user queries that are responded with an answer (as opposed to *no answer*) and the % of answers that are completely extracted from the given passage, *i.e.*, without any abstraction. We observe that: (*a*) algorithms that transit through states $\boldsymbol{ac}$ and $\boldsymbol{ss}$, *i.e.*, those that explicitly reason about query answerability and answer sentences, deem more questions as unanswerable, with the FLAN-UL2-20B assistant predicting thus more often than FALCON-40B, and (*b*) algorithms that search for relevant sentences ($\boldsymbol{ss}$) often copy document sentences verbatim in the answer, exhibiting the least abstraction.

Next we evaluate the faithfulness of the generated agent answers to the grounding document using two metrics: (*a*) the lexical precision of the answer with respect to the document after stopword removal and stemming, and (*b*) the WeCheck (Wu et al., 20223) factual consistency score of the same. Note that this evaluation only concerns actual answers and not the *no answer* responses, as we aim to exclusively measure hallucination with it. The results of Table 1 clearly indicate that both answerability classification ($\boldsymbol{ac}$) and answer sentence selection ($\boldsymbol{ss}$) improve agent faithfulness through SCOT prompting, and use of the FLAN-UL2-20B assistant leads to more faithful generation than using FALCON-40B in these steps. Overall, we observe an improvement of up to 16.8% over the simplest algorithm defined by the $\boldsymbol{uu} \to \boldsymbol{au}$ transition.

Finally, we take an LLM-as-a-reference approach to assessing the overall quality of our generated data as follows: We first prompt a high-performance MIXTRAL-8X7B-INSTRUCT-V0.1 model (MIXTRAL-I from here on) to generate agent utterances for all user queries in our synthetic data. MIXTRAL-I is given the queries along with their conversation histories, as generated by the original algorithm. We then evaluate the original agent utterances against the ones generated by MIXTRAL-I. We observed this evaluation strategy to be generally more reliable than asking MIXTRAL-I to judge generated utterances. In essence, we measure the extent to which our different processes behave like a high-performance instruction-tuned LLM.

Importantly, the presence of both answerable and unanswerable questions in our task calls for an evaluation protocol that incorporates the two individual classes and also assesses holistically on both. We measure lexical unigram precision and recall (after stopword removal and stemming) in the answerable class (as deemed by MIXTRAL-I) and compute a final F1-score; for the unanswerable class, a second F1-score is computed where precision and recall are both 1 if the agent response

Generator	State Transitions	Assistant	F_1-HM	F_1 (A)	F_1 (UA)
None (0-shot)	N/A	N/A	26.1±4.5%	54.0	17.2
HUMAN	N/A	N/A	46.6±5.5%	67.3	35.7
FALCON-40B	$uu \rightarrow au$	N/A	38.4±4.6%	67.9	26.8
	$\boldsymbol{uu \rightarrow ac \rightarrow au}$	None	41.0±2.9%	67.5	29.5
		FLAN-UL2-20B	**54.6***±2.5%	64.5	47.4
	$uu \rightarrow ac \rightarrow ss \rightarrow au$	None	45.1±2.0%	64.9	34.6
		FLAN-UL2-20B	52.4*±3.6%	63.7	44.6

Table 2: In-domain performance of LLAMA-2-13B-CHAT as the QA Agent on our seed demonstrations when prompted with various datasets. Asterisks (*) indicate improvement over human-annotated data.

indicates that the question is unanswerable, otherwise both are 0. We use the harmonic mean of the two F1-scores as our final evaluation metric to reward class-balanced performance.

The last three columns of Table 1 show the performances of all five algorithms. The three and four-step SCoT-prompted algorithms augmented with a FLAN-UL2-20B assistant have the best combined scores. On a closer look, the improvements from these approaches can be attributed to a big jump in performance in the unanswerable class, trading off much less accuracy in the answerable class. We also look at the related metric of answerability classification accuracy (Cls-Acc in Table 1) and observe a clear correlation with F1-score: algorithms with better overall performances are those that are better and more balanced at question answerability classification. These results crucially suggest that the primary source of hallucination in FALCON-40B in our data is misclassification, as it often produces answers to unanswerable questions.

4.2 Extrinsic Evaluation

4.2.1 Setup

Our extrinsic evaluation involves training QA agents with our generated data to produce responses to user queries in an ongoing conversation, and evaluating the agents on unseen test sets. We use two conversational QA datasets in our experiments: DOQA (Campos et al., 2020) and QUAC (Choi et al., 2018). DOQA (v2.1) has a training set of $1,037$ conversations containing $4,612$ dialogue turns; the test set has $1,200$ dialogues, with more than 4 turns on average per dialogue in three different domains: Cooking, Travel and Movies. For Wikipedia-based QUAC, we use the official dev set of $1,000$ conversations as our test set, and split the official train set of $11,567$ conversations into $10,567$ for training and $1,000$ for validation. QUAC has 7.2 turns per dialogue on average. We refer to the original papers for more detailed statistics.

We evaluate all generated data using both few-shot prompting and supervised fine-tuning of QA agents, as described in the next two sections. Our evaluation metric is the F1-score of §4.1.

4.2.2 Few-Shot Prompting

In few-shot evaluation, a trainee QA agent receives inference-time supervision in the form of in-context learning (ICL) demonstrations, which are sampled from our generated data. The agent is then asked to produce a response to a user query in an ongoing gold conversation based on a grounding document, similarly to state $\boldsymbol{au}$ of our state machine (§2). Given the limited amount of supervision that can be provided through ICL, we use an existing open-domain chat model LLAMA-2-13B-CHAT (Touvron et al., 2023) (LLAMA2-C henceforth) in our few-shot evaluation experiments.

Our first ICL evaluation re-uses the 6 seed demonstrations for an in-domain *roundtrip* assessment as follows: Given a 1000-dialogue synthetic dataset from §4.1 as the population of demonstrations, we run c evaluation cycles, each consisting of r rounds of evaluation. In each round, we uniformly sample (a) 2 document-conversation pairs from our synthetic dataset, and (b) one of the 32 user queries from the seed demonstrations along with its grounding document and conversation history. We then prompt LLAMA2-C with the synthetic demonstrations to generate an agent response for the seed query, and evaluate against the gold response. An evaluation cycle is complete when all r rounds in it have ended and the results have been averaged. Our final results consist of an average F1-score over the c cycles. Table 2 shows evaluation results for $c = 3$ and $r = 1000$. Standard deviations are reported as a $\%$ of the mean for comparability across methods. We first observe that all 2-shot results are better than the 0-shot results, indicating that LLAMA2-C can benefit from runtime demonstrations. Second, our methods that

Generator	State Transitions	Assistant	F_1-HM	F_1 (A)	F_1 (UA)
None (0-shot)	N/A	N/A	30.5±0.0%	28.7	32.6
HUMAN (Wikipedia)	N/A	N/A	46.0±0.6%	38.3	57.7
HUMAN (DOQA)	N/A	N/A	46.2±0.9%	49.6	43.2
FALCON-40B (Wikipedia)	$uu \rightarrow ac \rightarrow ss \rightarrow au$	FLAN-UL2-20B	**50.4***±0.9%	40.4	66.8

(a) **DOQA (Cooking) Results**

Generator	State Transitions	Assistant	F_1-HM	F_1 (A)	F_1 (UA)
None (0-shot)	N/A	N/A	39.2±0.0%	44.4	35.1
HUMAN (Wikipedia)	N/A	N/A	53.6±2.2%	51.9	55.4
HUMAN (QUAC)	N/A	N/A	50.1±0.7%	56.5	45.1
FALCON-40B (Wikipedia)	$uu \rightarrow ac \rightarrow ss \rightarrow au$	FLAN-UL2-20B	**56.5***±2.7%	52.4	61.3

(b) **QUAC Results**

Table 3: Performance of LLAMA-2-13B-CHAT as a QA Agent on two external benchmarks when few-shot prompted with various datasets. Data generated by our 4-step algorithm outperforms human-written demonstrations.

execute SCOT prompting with dedicated actions for question answerability classification and answer sentence selection, especially with a FLAN-UL2-20B assistant, perform the best. As in §4.1, they achieve a better balance between answerable and unanswerable class performance. The human demonstrations are sampled from our seed data so that the test query is from a different conversation. Interestingly, two of our methods outperform human demonstrations, likely due to the (much larger) synthetic datasets containing more topically similar passages with test instances.

Next we evaluate on the DOQA (Cooking) and QUAC test sets. The setup is similar as before, with one difference: instead of randomly sampling an instance from the test set in each round, we evaluate on all test instances to complete a cycle.[1] We report results from only the best-performing synthetic datasets in Table 3. The four-step algorithm with a FLAN-UL2-20B assistant produces the best synthetic data for ICL on both test sets, again outperforming our seed demonstrations. Interestingly, the synthetic data improves performance on the unanswerable class even over target domain human-labeled data, resulting in less hallucination and better overall results.

In summary, we observe in both in-domain and out-of-domain evaluation that SCOT prompting methods that leverage a FLAN-UL2-20B assistant to reduce hallucination perform the best for ICL.

4.2.3 Supervised Fine-Tuning (SFT)

To further examine the utility of our generated data, next we fine-tune a pre-trained LLM (FALCON-7B) on each dataset with QLORA (Dettmers et al., 2023), and evaluate on the test sets of DOQA and QUAC. Given the relatively high cost of SFT experiments involving our suite of five algorithms, we adopt the following two-stage process for evaluation: 1. Compare all five algorithms by (1a) training a FALCON-7B model on relatively small amounts of data from each and (1b) evaluating on validation sets; and 2. Assess the utility of the two best algorithms identified in step 1 more closely by (2a) generating more data with each to fine-tune a new FALCON-7B model and (2b) evaluating on test sets.

In our implementation of step 1 (details in Appendix B), the five 1000-conversation datasets of §4.1 are re-used as training data. The evaluation identifies the simplest $\boldsymbol{uu \rightarrow au}$ algorithm and the most advanced $\boldsymbol{uu \rightarrow ac \rightarrow ss \rightarrow au}$ with a FLAN-UL2-20B assistant as the two best algorithms on the two validation sets (see Table 5 of Appendix B). Interestingly, despite generating the least faithful conversations among our different algorithms (§4.1), and unlike in few-shot prompting (§4.2.2), the two-step algorithm performs strongly in SFT, indicating that SFT can be more robust to noisy training data than ICL as long as there is useful signal in it.

For step 2, we generate $10,000$ conversations with each of the above two algorithms. Table 4 presents a detailed comparison of performances on all four DOQA and QUAC test sets. First of all, our synthetic datasets demonstrate strong standalone performance (S_1 and S_2) on most test sets when compared to target domain training data. We also show cross-domain performances of DOQA and QUAC on each other for comparison, where we observe their training sets to lag well behind our synthetic data, which was also not generated specifically for any of the two domains.

A closer look at the answerable (A) and unan-

[1]For QUAC training, we are able to use only one conversation – often longer than the other datasets – as demonstration before the input length reaches the limit for FALCON-40B.

Training Set	Evaluation Sets											
	DoQA Cooking			DoQA Movies			DoQA Travel			QuAC		
	A	UA	HM	A	UA	HM	A	UA	HM	A	UA	HM
D: DoQA Gold	26.4	60.1	36.6	20.8	71.1	32.2	25.1	63.3	35.9	22.1	73.1	34.0
Q: QuAC Gold	10.2	90.6	18.3	7.1	95.5	13.2	8.1	97.2	15.0	35.0	85.6	49.7
S_1: $uu \rightarrow au$	22.8	65.2	33.8	19.0	59.4	28.8	19.0	67.0	29.6	38.6	43.1	40.7
S_2: $uu \rightarrow ac \rightarrow ss \rightarrow au$	19.2	85.8	31.3	14.2	90.4	24.5	18.1	87.5	29.9	28.4	82.8	42.3
S_3: 50% $S_1 \cup$ 50% S_2	22.5	71.9	34.3	18.4	77.1	29.7	18.9	78.0	30.5	37.2	49.3	42.4
D augmented w/ S_3	40.7	66.3	**50.4**	34.2	66.4	**45.1**	39.9	66.2	**49.8**		–	
Q augmented w/ S_3		–			–			–		49.8	75.0	**59.8**

Table 4: F_1-scores of FALCON-7B models fine-tuned (with QLoRA) on various training datasets (§4.2.3).

swerable (UA) class performances reveals the same disparity between S_1 and S_2 as before (*e.g.*, in Table 1): S_1 performs better on answerable queries and S_2 on unanswerable ones. We therefore also consider a 50:50 mixture of the two as a third training set (10,000 conversations), termed S_3 in the table, to find out how well they complement each other. The results confirm the strength of their combination, as the mixture outperforms the individual datasets on all test sets.

Our final SFT evaluation measures the ability of our synthetic data to augment in-domain gold data, for which we fine-tune the model trained on S_3 – the best synthetic dataset – further on target domain training examples separately for DoQA and QuAC. As indicated by the results in Table 4 (last 2 rows), this augmentation provides a strong boost to results in both domains over training only on gold data, with improvements ranging from 10.1% to 13.9%. These remarkable results showcase the out-of-domain utility of our synthetic data derived from only 6 simple Wikipedia-based demonstrations. We discuss other methods of augmentation that we experimented with in Appendix B.

5 Related Work

Chain-of-thought (CoT) prompting (Wei et al., 2022) showed that LLMs can be few-shot prompted (Brown et al., 2020) to produce intermediate reasoning steps, or "thoughts", which can improve their performance. Prompts were created manually and were application-specific (see (Wei et al., 2022, Appendix G)). Chain-of-verification (CoVe) (Dhuliawala et al., 2023) adopted a structured approach to CoT for answer verification and error correction, splitting the thought process into verification planning and execution steps that utilize their own prompts, finally producing a rewritten response with reduced hallucination. Also related is the recitation-augmented CoT approach of Sun et al. (2023b), which recites related knowledge stored in the LLM's parameters before answering a question. Other structured CoT prompting work include ReAct (Yao et al., 2023) and ReWoo (Xu et al., 2023a), which leverage LLM-powered agents to solve complex tasks through interleaved steps, with use of external tools such as search engines.

The proposed approach draws inspiration from the above studies, but aims to solve the complex task of multi-turn content-grounded conversation generation, focusing on the key associated requirements of determining question answerability and knowledge selection. More importantly, we produce synthetic data that can train agents for direct inference, removing the need for computationally expensive CoT prompting during inference.

The CoT precursor Scratchpad (Nye et al., 2021) showed that models can also be fine-tuned to execute thought processes in order to solve a task in a step-by-step manner, and the benefits of generating synthetic CoT data for training. It did not, however, use prompting and was limited to relatively close-ended problems. Constitutional AI (Bai et al., 2022) used a two-step CoT process (critique and revise) to generate corrected responses. Like this work, and unlike Scratchpad, it removed the CoT and kept only the final response to generate the synthetic data. Unlike our work, Constitutional AI focuses on reducing harmfulness and uses this step as part of a bigger process that includes reinforcement learning.

Finally, it is worth noting that many prior works relied on closed-source models such as GPT-3.5[2], PaLM (Chowdhery et al., 2023), Codex (Chen et al., 2021) and LaMDA (Thoppilan et al., 2022), and are thus not reproducible. Here we only use open-source models whose weights are publicly available, both for generation: FALCON (Penedo et al., 2023) and FLAN (Tay et al., 2022), and for eval-

[2]Which is being deprecated and is no longer accessible https://platform.openai.com/docs/deprecations/instructgpt-models

uation: MIXTRAL (Jiang et al., 2024), LLAMA-2 (Touvron et al., 2023) and FALCON.

6 Conclusion

We introduce a structured chain-of-thought (SCoT) prompting approach to generating multi-turn content-grounded conversations and empirically show that high-quality synthetic data can be produced from only six human-written seed conversations. Designated states for hallucination mitigation and the use of supporting tools enable our methods to generate agent utterances that are highly faithful to grounding documents. Used as training data, our generated conversations train high-performance models as evaluated on out-of-domain test sets, successfully augmenting target-domain human-labeled data. Future work will explore more complex conversational settings, *e.g.*, multi-document grounding and response generation for harder, more ambiguous user utterances.

Limitations

The goal of this work is to mitigate hallucination in pre-trained LLMs through structured CoT prompting and LM augmentation, improving overall generation quality. Even though we provide ample empirical evidence of successful hallucination mitigation through both intrinsic and extrinsic evaluation, given the high cost of manual labor involved in evaluating long multi-turn conversations generated by a large number of algorithms, we only perform automatic evaluation, relying on cutting-edge LLMs and factual consistency checking models. While the use of only 6 seed conversations was sufficient to demonstrate the strength of our approach, our generated data could be more diverse and train even better models if more demonstrations were utilized.

References

Vaibhav Adlakha, Shehzaad Dhuliawala, Kaheer Suleman, Harm de Vries, and Siva Reddy. 2022. TopiOCQA: Open-domain Conversational Question Answering with Topic Switching. *Transactions of the ACL*.

Leonard Adolphs, Kurt Shuster, Jack Urbanek, Arthur Szlam, and Jason Weston. 2022. Reason first, then respond: Modular Generation for Knowledge-infused Dialogue. In *Findings of EMNLP*.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. The Falcon Series of Open Language Models. *arXiv preprint arXiv:2311.16867*.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional AI: Harmlessness from AI Feedback. *arXiv preprint arXiv:2212.08073*.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity. In *IJCNLP-ACL*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language Models are Few-Shot Learners. In *NeurIPS*.

Jon Ander Campos, Arantxa Otegi, Aitor Soroa, Jan Deriu, Mark Cieliebak, and Eneko Agirre. 2020. DoQA - Accessing Domain-Specific FAQs via Conversational QA. In *ACL*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question Answering in Context. In *EMNLP*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient Finetuning of Quantized LLMs. In *NeurIPS*.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2023. Chain-of-Verification Reduces Hallucination in Large Language Models. *arXiv preprint arXiv:2309.11495*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The Curious Case of Neural Text Degeneration. In *ICLR*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and

Ting Liu. 2023. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. *arXiv preprint arXiv:2311.05232*.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of Experts. *arXiv preprint arXiv:2401.04088*.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On Faithfulness and Factuality in Abstractive Summarization. In *ACL*.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. 2021. Show Your Work: Scratchpads for Intermediate Computation with Language Models. *arXiv preprint arXiv:2112.00114*.

OpenAI. 2023. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*.

Yifu Qiu, Yftah Ziser, Anna Korhonen, Edoardo Ponti, and Shay B Cohen. 2023. Detecting and Mitigating Hallucinations in Multilingual Summarisation. In *EMNLP*.

Siva Reddy, Danqi Chen, , and Christopher Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the ACL*.

Mobashir Sadat, Zhengyu Zhou, Lukas Lange, Jun Araki, Arsalan Gundroo, Bingqing Wang, Rakesh R Menon, Md Rizwan Parvez, and Zhe Feng. 2023. DelucionQA: Detecting Hallucinations in Domain-specific Question Answering. In *Findings of EMNLP*.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language Models Can Teach Themselves to Use Tools. In *NeurIPS*.

Weiwei Sun, Pengjie Ren, and Zhaochun Ren. 2023a. Generative Knowledge Selection for Knowledge-Grounded Dialogues. In *Findings of ACL*.

Zhiqing Sun, Xuezhi Wang, Yi Tay, Yiming Yang, and Denny Zhou. 2023b. Recitation-Augmented Language Models. In *ICLR*.

Yi Tay, Mostafa Dehghani, Vinh Q Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Dara Bahri, Tal Schuster, Steven Zheng, et al. 2022. Ul2: Unifying language learning paradigms. In *The Eleventh International Conference on Learning Representations*.

Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Siamak Shakeri, Dara Bahri, Tal Schuster, Huaixiu Steven Zheng, Denny Zhou, Neil Houlsby, and Donald Metzler. 2023. UL2: Unifying Language Learning Paradigms. In *ICLR*.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, et al. 2022. LaMDA: Language Models for Dialog Applications. *arXiv preprint arXiv:2201.08239*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv preprint arXiv:2307.09288*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *NeurIPS*.

Wenhao Wu, Wei Li, Xinyan Xiao, Jiachen Liu, Sujian Li, and Yajuan Lv. 20223. WeCheck: Strong Factual Consistency Checker via Weakly Supervised Learning. In *ACL*.

Binfeng Xu, Zhiyuan Peng, Bowen Lei, Subhabrata Mukherjee, Yuchen Liu, and Dongkuan Xu. 2023a. ReWOO: Decoupling Reasoning from Observations for Efficient Augmented Language Models. *arXiv preprint arXiv:2305.18323*.

Weijia Xu, Sweta Agrawal, Eleftheria Briakou, Marianna J. Martindale, and Marine Carpuat. 2023b. Understanding and Detecting Hallucinations in Neural Machine Translation via Model Introspection. *Transactions of the ACL*.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024. Hallucination is Inevitable: An Innate Limitation of Large Language Models. *arXiv preprint arXiv:2401.11817*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In *ICLR*.

A Prompts

A.1 Prompts for the Remaining States

In Figure 3, we showed the prompts used in agent utterance generation and answer sentence selection. Figures 4, 5, 6 and 7 illustrate our remaining prompts used at various stages of the different algorithms.

A.2 Details of ICL Demonstrations

In states $\boldsymbol{uu}$ and $\boldsymbol{au}$, we use two full document-grounded conversations as demonstrations. In question answerability classification ($\boldsymbol{ac}$), we use 3 positive and 3 negative exemplars for the FLAN-UL2-20B assistant, and 2 positive and 2 negative exemplars for FALCON-40B; for the latter, a larger number of demonstrations often exceeds the maximum input length limit. This is an example of an advantage that comes with LLM augmentation (§1), where a different tool can execute actions and/or leverage resources that the primary LLM cannot. In sentence search ($\boldsymbol{ss}$) we use 6 exemplars for the FLAN-UL2-20B assistant; for FALCON-40B, we are only able to use 3.

```
Here is a document broken down into its title and text:
Title: Harry Potter
Text: Harry Potter is a series of …
EoD
The following is a conversation between a User and an Agent, where the User asks a
series of questions about the topic of the document and the Agent tries to answer those
questions from the document:
User says: Tell me about Harry Potter. EoU
Agent says: Harry Potter is a … EoA
…
User says: Are there any movies based on the series? EoU

Here is a document broken down into its title and text:
Title: …
Text: …
EoD
The following is a conversation between a User and an Agent, where the User asks a
series of questions about the topic of the document and the Agent tries to answer those
questions from the document:
User says: … EoU
Agent says: … EoA
…
User says: …
```

Figure 4: Prompt for a pre-trained LLM in state $\boldsymbol{uu}$: user utterance generation (§3).

```
You are given a document with its title and text. You are also given an ongoing
conversation between a User and an Agent. Determine if the document has enough
information to answer the last User query in the conversation.

Document:
Title: Harry Potter
S1: Harry Potter is a series of …
S2: The novels chronicle the life of …
S3: The main story arc concerns …
…
EoD
Conversation:
User says: Tell me about Harry Potter
Agent says: Harry Potter is a …
User says: What happens in the story?
Does the document have enough information to answer this last User query? Yes/No: Yes

Document:
Title: …
S1: …
…
EoD
Conversation:
User says: …
Agent says: …
…
User says: …
Does the document have enough information to answer this last User query? Yes/No:
```

Figure 5: Prompt for an instruction-following LLM assistant in state $\boldsymbol{ac}$: question answerability classification (§3).

```
Here is a document broken down into its title and text:
Title: Harry Potter
Text: …
EoD
Here is an ongoing conversation between a User and an Agent:
User says: Tell me about Harry Potter
Agent says: Harry Potter is a …
User says: What happens in the story?
Does the document answer this last User query? Yes/No: Yes

Here is a document broken down into its title and text:
Title: …
Text: …
EoD
Here is an ongoing conversation between a User and an Agent:
User says: …
Agent says: …
…
User says: …
Does the document answer this last User query? Yes/No:
```

Figure 6: Prompt for a pre-trained LLM (*i.e.*, no assistant) in state $\boldsymbol{ac}$: question answerability classification (§3).

```
Here is a document broken down into its title and text:
Title: Harry Potter
Text: Harry Potter is a …
EoD
Here is an ongoing conversation between a User and an Agent:
User says: Tell me about Harry Potter
Agent says: Harry Potter is a …
User says: What happens in the story?
The following sentences from the above document contain an
answer to this last User query: The main story arc concerns …

Here is a document broken down into its title and text:
Title: …
Text: …
EoD
Here is an ongoing conversation between a User and an Agent:
User says: …
Agent says: …
…
User says: …
The following sentences from the above document contain an
answer to this last User query:
```

Figure 7: Prompt for a pre-trained LLM (*i.e.*, no assistant) in state ss (§3).

B SFT Details

In this section, we provide details on the instruction and input format used for our SFT experiments. We also present results from all steps from our workflow described in Section 4.2.3.

B.1 SFT Instruction Prompt

We use the instruction and input format described in Listing 1 when fine-tuning the FALCON-7B model (with QLORA).

```
INSTRUCTION = "Given the document
    ↪ and the current conversation
    ↪ between a user and an agent,
    ↪ your task is to generate the
    ↪ next response from the agent.

While generating the agent response
    ↪ you should: Determine if
    ↪ agent response needs
    ↪ information from document.
(a) If yes, generate agent response
    ↪ using only precise
    ↪ information present in the
    ↪ document.
(b) If not, generate CANNOTANSWER. "

input = INSTRUCTION + " \n\n "
input += "Text: " + DOC_TITLE + " :
    ↪ " + DOC_TEXT + "\n\n"
input += f"Input: "

for utt, speaker in zip(dialog["
    ↪ utterances"],dialog["speakers
    ↪ "]):
    input += f"User: " if speaker ==
        ↪  "user" else "Agent: "
    input += f"{utt.strip()} "

input = input.rstrip()
input += f"\n\nOutput:"
```

Listing 1: Instruction and Input format

We use 128 as the maximum number of tokens for generation during our SFT experiments. We filter out few instances from the DOQA and QUAC datasets based on length to ensure that each input instance is less than 1920 tokens (2048 model max input length - 128 max output tokens for generation). We create input instances as mentioned above and use the following check:

```
tokenizer = AutoTokenizer.
    ↪ from_pretrained("tiiuae/falcon-40
    ↪ b")
if len(tokenizer(input)['input_ids']) >
    ↪ (2048 - 128):
    return False
else:
    return True
```

This filtering reduces size of the QUAC data as follows: For the train set, the number of instances is reduced from 76129 to 75194; for the dev set: from 7439 to 7332; and for the test set: from 7354 to 7069 instances.

We fine-tune the FALCON-7B model[3] (with QLORA) across all training data with 2 Tesla A100 GPUs. We use the following settings across all training runs:

- quantization method is set to 'fp4'
- LoRA rank, alpha, dropout and target modules are set to 8, 32, 0.1 and ["query_key_value"]
- Batch size and learning rate are set to 1 and $4.0e-4$
- DeepSpeedFusedAdam is used as the optimizer with weight decay = 0.1, betas = [0.9, 0.95] and eps = $1e-10$
- warmup steps are set to 1000 steps with a linear learning rate schedule.

We save checkpoints at periodic intervals during training and evaluate the models on the dev dataset to select the checkpoint that achieves the best validation performance for the final evaluation on the test set. For synthetic data, the models are trained for 15 epochs. For GOLD data, the models are trained for 10 epochs and the augmented models 'D augmented w/ S3' and 'Q augmented w/ S3' are fine-tuned for 5 epochs.

B.2 Identifying Best Synthetic Data

As described in Section 4.2.3, we adopt a step-by-step workflow to identify the best synthetic data for the best downstream task performance. In this section, we present the details for each step and share our results from these steps on our two downstream tasks: DOQA and QUAC.

Training Set	A	UA	HM
DoQA			
D: DoQA GOLD	24.5	65.5	35.7
$uu \to au$	22.4	55.4	**31.9**
$uu \to ac \to ss \to au$ with a FLAN-UL2-20B Assistant	12.8	95.6	22.5
$uu \to ac \to ss \to au$ with no Assistant	18.9	53.0	27.9
$uu \to ac \to au$ with a FLAN-UL2-20B Assistant	11.7	96.0	20.8
$uu \to ac \to au$ with no Assistant	17.6	73.5	28.4
QuAC			
Q: QuAC GOLD	25.2	83.0	38.7
$uu \to au$	25.8	43.0	32.2
$uu \to ac \to ss \to au$ with a FLAN-UL2-20B Assistant	21.7	71.5	**33.3**
$uu \to ac \to au$ with no Assistant	26.1	41.0	31.9
$uu \to ac \to au$ with a FLAN-UL2-20B Assistant	15.4	88.5	26.2
$uu \to ac \to ss \to au$ with no Assistant	16.6	54.0	25.4

Table 5: Performance of FALCON-7B models fine-tuned (with QLORA) on 1,000 synthetic conversations on DoQA Cooking dev set and QuAC dev set.

B.2.1 Step 1: Identifying the best algorithms

In Step 1, we use the synthetic conversations generated on 1,000 Wikipedia passages from Section 1 using various state transitions and fine-tune FALCON-7B models (with QLORA) on them. In Table 5, we present results of our SFT evaluation using models fine-tuned only on 1000 synthetic conversations. For evaluation, we use the official DoQA Cooking dev set and QuAC dev set.

We observe that among the synthetic datasets, "$uu \to au$" model performs best on the A class. For the UA class, two approaches: "$uu \to ac \to ss \to au$ with a FLAN-UL2-20B Assistant" and "$uu \to ac \to au$ with a FLAN-UL2-20B Assistant" perform quite well, with the former performing better on the A class. Hence, for our next stage of SFT experiments with larger synthetic data, we choose the "$uu \to au$" and the "$uu \to ac \to ss \to au$ with a FLAN-UL2-20B Assistant" settings as our preferred synthetic data generation methods for the DoQA and QuAC downstream tasks.

B.2.2 Step 2: Identifying the best synthetic data mixture

In Step 2, we fine-tune models on $10,000$ synthetic conversations ($10x$ from Step 1). In Table 6, we present results of our SFT evaluation for these models. We use the DoQA and QuAC dev set as before for evaluation. We observe that with $10,000$ conversations, "$S1$: $uu \to au$" model performs similar to $1,000$ conversations in Table 5 for both DoQA and QuAC. For DoQA, we notice a significant improvement in the performance of "$S2$: $uu \to ac \to ss \to au$ with a FLAN-UL2-20B Assistant" model (7.5% absolute gain on HM metric).

In addition to improvements from larger synthetic data size, the most significant finding is with the $S1 \cup S2$ approach where we fine-tune FALCON-7B model with a $50 - 50$ ratio of $S1$ and $S2$. We observe that models trained with $S1 \cup S2$ approach outperform models trained with $S1$ or $S2$ individually. This showcases that the synthetic data generated using our different approaches complements each other and improves the performance on the downstream task.

B.2.3 Step 3: Further improvement using data augmentation

We explore several other approaches for utilizing the synthetic data, such as:

- augmenting the synthetic data with GOLD data and fine-tuning from scratch
- further fine-tuning the model with GOLD data already trained on synthetic data, and
- further fine-tuning the model with synthetic data already trained on GOLD data

We use DoQA dataset for running these additional experiments and report our results in Table 7. We observe that the data augmentation improves performance, even when using only 20% of the synthetic data mixture $S3$. Using more synthetic data for augmentation does not yield additional gain in performance. However, this requires retraining the models from scratch, which may be expensive based on model and dataset size. The $S3$ on D model, achieved by further fine-tuning the model (already trained on DoQA GOLD data) with $S3$ data does not improve performance. This

[3] tiiuae/falcon-7b

Training Set	A	UA	HM
DoQA			
D: DoQA GOLD	24.5	65.5	35.7
$S1$: $\boldsymbol{uu} \rightarrow \boldsymbol{au}$	22.4	59.4	32.5
$S2$: $\boldsymbol{uu} \rightarrow \boldsymbol{ac} \rightarrow \boldsymbol{ss} \rightarrow \boldsymbol{au}$ with a FLAN-UL2-20B Assistant	18.7	90.0	31.0
$S1 \cup S2$	22.1	75.1	34.2
QuAC			
Q: QuAC GOLD	25.2	83.0	38.7
$S1$: $\boldsymbol{uu} \rightarrow \boldsymbol{au}$	24.6	55.4	34.1
$S2$: $\boldsymbol{uu} \rightarrow \boldsymbol{ac} \rightarrow \boldsymbol{ss} \rightarrow \boldsymbol{au}$ with a FLAN-UL2-20B Assistant	19.3	84.4	31.4
$S1 \cup S2$	26.2	58.8	36.2

Table 6: Performance of FALCON-7B models fine-tuned (with QLORA) on 10,000 synthetic conversations on DoQA Cooking dev and QuAC dev sets.

is expected since our synthetic data is not related to the GOLD data; hence the drop in performance.

The D on $S3$ model, *aka* Alignment model, is achieved by using the model fine-tuned on our synthetic data mixture $S3$ and further fine-tuning it with the DoQA GOLD data outperforms the models trained from scratch with data augmentation. This shows that our synthetic data is of good quality and can be used to train a good initial model, which can be used for further domain alignment with GOLD data available for the domain.

Training Set	A	UA	HM
D: DoQA GOLD	24.5	65.5	35.7
$S1$: $\boldsymbol{uu} \rightarrow \boldsymbol{au}$	22.4	59.4	32.5
$S2$: $\boldsymbol{uu} \rightarrow \boldsymbol{ac} \rightarrow \boldsymbol{ss} \rightarrow \boldsymbol{au}$ w/ FLAN-UL2-20B Asst.	18.7	90.0	31.0
$S3$: $S1 \cup S2$	22.1	75.1	34.2
$D \cup S3(20\%)$	33.0	75.9	46.0
$D \cup S3(40\%)$	35.6	62.3	45.3
$D \cup S3(60\%)$	36.2	63.5	46.1
$D \cup S3(80\%)$	34.8	68.7	46.2
$D \cup S3(100\%)$	34.9	67.9	46.1
$S3$ on D model	16.2	89.6	27.5
D on $S3$ model	37.7	66.3	48.1

Table 7: Performance of FALCON-7B models fine-tuned (with QLORA) on 10,000 synthetic conversations on DoQA Cooking dev set.