

Human-Centered Interactive Information Seeking

Zequ Wu

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington
2024

Reading Committee:
Mari Ostendorf, Co-Chair
Hannaneh Hajishirzi, Co-Chair
Yejin Choi

Program Authorized to Offer Degree:
Electrical and Computer Engineering

© Copyright 2024

Zequi Wu

University of Washington

Abstract

Human-Centered Interactive Information Seeking

Zeqiu Wu

Co-chairs of the Supervisory Committee:

Professor Mari Ostendorf

Electrical and Computer Engineering

Associate Professor Hannaneh Hajishirzi

Computer Science and Engineering

The rapidly increasing volume of accessible information in the digital age underscores the need for developing automated models to facilitate our daily information-seeking tasks. Moreover, as humans often have exploratory information needs and do not always come up with very descriptive initial queries, it is important to build an AI agent that can converse and collaborate with human users to find the information they are interested in. Upon receiving and interpreting a user’s request in a conversation, the AI agent should compose a response grounded in relevant information from a knowledge source.

In this thesis, we introduce our proposed solutions to three pivotal challenges in this interactive process. First, identifying relevant knowledge to be used in conversational systems that are grounded in long documents is critical to effective response generation. We introduce a knowledge identification model DIALKI, that leverages the document structure to provide dialogue-contextualized passage encodings and better locate knowledge relevant to the conversation. An auxiliary loss captures the history of dialogue-document connections. We demonstrate the effectiveness of DIALKI on two document-grounded conversational datasets and provide analyses showing generalization to unseen documents and long dialogue contexts. When the knowledge source contains millions or more documents or passages, passage retrieval is performed before response generation. As it is often not practical to re-train well-established retrievers like a search engine

to handle conversational queries specifically, we develop a query rewriting model CONQRR that rewrites a question that depends on dialogue context in the context into a standalone question. It is trained with a novel reward function to directly optimize towards retrieval using reinforcement learning and can be adapted to any off-the-shelf retriever. CONQRR achieves state-of-the-art results on a recent open-domain conversational question answering dataset containing conversations from three different sources, and it is effective for two different off-the-shelf retrievers. Our extensive analysis also shows the robustness of CONQRR to out-of-domain dialogues as well as to a scenario where no query rewriting supervision is available.

The second challenge comes from the need for AI agents to collaborate with users during the information-seeking process. To facilitate this line of research, we present INSCIT, a dataset for **I**nformation-**S**eeking **C**onversations with mixed-initiative **I**nteractions. It contains 805 human-human conversations where the agent searches over Wikipedia and either directly answers, asks for clarification, or provides relevant information to address user queries. Along with the data, we propose two subtasks, evidence passage identification and response generation, as well as a human evaluation protocol to assess model performance. We develop two strong baselines. Both of them significantly underperform humans, suggesting ample room for improvement in future studies.

Finally, to teach an AI agent to generate responses that can provide the maximized utility to the user, we propose to use fine-grained human feedback (e.g., which sentence is false, which sub-sentence is irrelevant) as an explicit training signal. We introduce FINE-GRAINED RLHF, a framework that enables training and learning from reward functions that are learnt from such fine-grained human feedback. Our experiments illustrate how learning with such reward functions leads to improved performance in response generation for information queries, supported by both automatic and human evaluation. Additionally, we show that LM behaviors can be customized using different combinations of fine-grained reward models.

Acknowledgements

I would like to express my deepest gratitude to my advisors, Mari Ostendorf and Hannaneh Hajishirzi. Their invaluable guidance and insight have been crucial throughout my PhD studies. They have shaped my ability to approach problems with a critical mindset and articulate my ideas with precision. Their consistent support and patience have allowed me to explore various research directions, and this journey would not have been possible without their mentorship.

I am fortunate to also have Yejin Choi and Aylin Caliskan to serve on my thesis committee. I appreciate their constructive feedback and valuable suggestions that significantly help improve this thesis.

I am proud to be a part of the TIAL Lab and H2 Lab at UW. Both labs provide incredibly supportive, friendly and vivid research environments. I extend my gratitude to all lab members and alumni with whom I've interacted over the past five years. Special thanks to Yi Luan, Hao Cheng, Rik Koncel-Kedziorski and Prithviraj Ammanabrolu (listed in the order I met them) for their support and assistance during challenging times. My appreciation also goes out to the entire UWNLP research community.

This thesis could not have been completed without the help of my research collaborators, particularly Yushi Hu and Roy Lu, whose significant contributions have been vital to some of the work in this thesis. I wish to express my gratitude to my internship mentor, Michel Galley at Microsoft Research. He was not only pivotal in helping me publish my first two PhD papers, but more importantly, he played a crucial role in teaching me on how to excel in research. I also want to thank Gaurav Singh Tomar, my mentor at Google Research, for his immense support during and following my internship.

Beyond research, I am blessed to have a diverse group of friends in Seattle who provide me with companionship and entertainment. I would especially like to express my gratitude to Tingting Zhao, Nayon Park, and Ather Sharif for their consistent care and attention towards me.

I am deeply grateful to my parents and family for their endless love and support from the moment of my birth. They have instilled in me the values of gratitude, resilience, and humility. Additionally, they have given me an unparalleled level of freedom and a liberal home environment, a rare privilege in an Asian family, empowering me to pursue my life's passions without any regrets. Finally, I extend my heartfelt thanks to my fiancé, Remy Wang, who cherishes me just as I am and has profoundly influenced me in numerous ways. His perspective has especially shaped my understanding of my role as a researcher and the world as a wondrous place to explore. I am looking forward to seeing the future with you.

Contents

1	Introduction	19
1.1	Knowledge Retrieval and Identification in Conversations	20
1.2	Interactive and Collaborative Information Seeking	21
1.3	Response Generation Informed by Human-centered Objectives	22
1.4	Thesis Overview	22
2	Background	25
2.1	Grounded Dialogue Systems	25
2.1.1	Social Chats	26
2.1.2	Information-Seeking Dialogues	26
2.1.3	Task-Oriented Dialogues	27
2.2	Document-Grounded Information-Seeking Dialogues	28
2.2.1	Evaluation	28
2.2.2	Models	30
2.3	Learning from Human Feedback	31
2.3.1	Reinforcement Learning (RL) for NLP	31
2.3.2	Reinforcement Learning from Human Feedback (RLHF)	32
2.3.3	Non-RL Learning from Human Feedback in NLP	32
3	DIALKI: Knowledge Identification in Conversational Systems through Dialogue-Document Contextualization	35
3.1	Introduction	35

3.2	Method	37
3.2.1	Encoding Dialog Context and Knowledge	38
3.2.2	DIALKI with Multi-Task Learning	39
3.2.3	Training and Inference	41
3.3	Experiment	42
3.3.1	Datasets and Evaluation Metrics	42
3.3.2	Implementation Details	43
3.3.3	Compared Systems	44
3.3.4	Quantitative Results	45
3.3.5	Analysis	48
3.4	Summary	50
4	CONQRR: Conversational Query Rewriting for Retrieval with Reinforcement Learning	53
4.1	Introduction	53
4.2	Approach	55
4.2.1	T5QR	55
4.2.2	CONQRR	56
4.2.3	Retriever Models	59
4.3	Experiment	59
4.3.1	Compared Systems	62
4.3.2	Quantitative Results	62
4.3.3	Analysis	63
4.4	Summary	69
5	INSCIT: Information-Seeking Conversations with Mixed-Initiative Interactions	73
5.1	Introduction	73
5.2	Task Formulations	75
5.3	Data: INSCIT	76
5.3.1	Data Collection Pipeline	77

5.3.2	Quality Control	78
5.3.3	Data Analysis	79
5.4	Experiment Setup	83
5.4.1	Systems	83
5.4.2	Evaluation	85
5.5	Experiment Results	86
5.5.1	Quantitative Results	86
5.5.2	Analysis	88
5.6	Ethical Considerations for Dataset Collection	90
5.7	Summary	91
6	Fine-Grained Human Feedback Gives Better Rewards for Language Model Training	93
6.1	Introduction	93
6.2	FINE-GRAINED RLHF	96
6.3	Task 1: Detoxification	97
6.3.1	Experiments	98
6.4	Task 2: Long-Form Question Answering (QA)	99
6.4.1	QA-FEEDBACK: Long Form QA with Human Feedback	100
6.4.2	Fine-Grained Reward Models	101
6.4.3	Experimental Setup	103
6.4.4	Main Results	104
6.4.5	LM Customization with FINE-GRAINED RLHF	105
6.4.6	Analysis	106
6.5	Discussion	108
6.5.1	Broader Impacts	108
6.5.2	Limitations	109
6.6	Summary	109

7	Conclusion	111
7.1	Summary and Contributions	111
7.1.1	Knowledge Retrieval and Identification in Conversations	111
7.1.2	Interactive and Collaborative Information Seeking	112
7.1.3	Response Generation Informed by Human-centered Objectives	112
7.2	Future Directions	113
A	Additional Examples and Training Details for FINE-GRAINED RLHF	137
A.1	Algorithm and Training Details of FINE-GRAINED RLHF	137
A.1.1	PPO with Fine-Grained Rewards	137
A.1.2	Implementation Details	137
A.2	Long-Form QA Data and Human Feedback Annotation	139
A.2.1	Human Feedback Annotation	139
A.2.2	Analysis of Collected Fine-Grained Feedback	140
A.3	Long-Form QA Reward Model Training Details	141
A.4	Qualitative Examples for Long-Form QA	143
A.4.1	Examples on LM Customization	143
A.4.2	Examples on LM Errors	147
A.4.3	Examples on Reward Model Ablation	150
A.4.4	Comparison with ChatGPT responses	152

List of Figures

3.1	In a document-grounded conversation, <i>knowledge identification</i> targets to locate a knowledge string within a long document to assist the agent in addressing the current user query.	35
3.2	The overview of DIALKI. Each document is divided into passages. We apply BERT and a knowledge contextualization mechanism to obtain dialogue context and knowledge representations (left), for performing both next (main) and history (auxiliary) turn knowledge identification tasks (right). For each turn, DIALKI identifies knowledge by selecting the relevant passage as well as the begin/end spans in the passage.	37
3.3	Similarity between \mathbf{z} and the latest 4 history turn representations (i.e., $\mathbf{u}_1 \dots \mathbf{u}_4$ from left to right) on Doc2Dial (left) and WoW (right).	48
3.4	EM versus the length of dialogue context (# previous turns) or document (# tokens).	49
4.1	A CQA agent rewrites the current user question into a more effective one (in orange) for the given <i>off-the-shelf</i> retriever to find the most relevant passage.	53
4.2	Our CONQRR framework. Yellow and blue arrows mark the flow of CE (unused when $\alpha = 1.0$) and RL loss calculation, respectively. During inference, only q (dashed border) is generated as the final rewrite.	56
4.3	MRR on QReCC versus the percentage of QR supervision used for training, with DE as the retriever.	64
4.4	MRR versus the number of questions in the dialogue context, with DE as the retriever.	66
4.5	Evaluation scores on QReCC for T5QR w/ or w/o brevity penalty and DIALKI (mix), with DE as the retriever. Recall scores (R@k) are divided by 100.	66

5.1	INSCIT examples show that user queries can often be under-specified and require clarification (left), or have no direct answer but where providing relevant information may fulfill users’ information needs (right).	73
5.2	Each conversation is annotated in a series of user → agent → validator tasks. One worker is dedicated to each user/validator task but two workers work in parallel on the agent turn annotation (see discussion in § 5.3.1).	76
5.3	Left: seed document topic category breakdown ($D \rightarrow$ category <i>depth</i>). Middle and right: treemaps of top 7 (and other: MISC) first tokens in user turns from conversations under “food & drink” and “historical events” topic categories. <i>For each figure, the size of each colored area is proportional to its percentage in the data.</i>	80
5.4	Human evaluation on system comparison for all dimensions: evidence utility (EU), factual consistency (FC), coherence (CO) and response comprehensiveness (COMP). Win/lose refers to DIALKI+FiD.	88
5.5	PI-F1 and RG-F1 scores by reference response strategy (direct answer, clarification, relevant answer) on the one-strategy test subset, excluding instances where two references differed in strategy.	89
6.1	Comparison of (a) RL with human preference and (b) our FINE-GRAINED RLHF on long-form QA. Different from (a), which collects human preferences on the overall quality of LM outputs, we ask annotators to mark which part of an output contains what type(s) of errors. We train a fine-grained reward model for each type of error and optimize LM against these reward models. In this example, we provide a relevance reward and a factuality reward after each sentence is generated. There is also a holistic information completeness reward after the whole text is generated.	94
6.2	Curves of toxicity and perplexity on the dev set vs. training steps.	99
6.3	Human evaluation on rel. (left) and fact. (right) error, measured by % of sub-sentences that contain the error type (↓).	104
6.4	Dynamics of each type of reward during training (reward vs. training steps). All rewards are z-normalized.	106

A.1	Fine-grained feedback annotation interface.	143
A.2	Error types in the fine-grained feedback annotation interface.	144
A.3	Preference feedback annotation interface. The task input is omitted (same as in Figure A.1).	144
A.4	Fine-grained feedback annotation instructions.	145

List of Tables

3.1	Evaluation results on the Doc2Dial test set.	45
3.2	Evaluation results of WoW test sets.	46
3.3	Ablation results on Doc2Dial and WoW dev sets.	46
3.4	Response generation results on Doc2Dial dev set. KI stands for Knowledge Identification.	47
3.5	Sample generated responses from BART with the full grounding document (baseline) or the predicted grounding span and passage by DIALKI (ours) as the additional input to the dialogue context.	47
3.6	Passage prediction accuracy on dev sets.	48
4.1	Passage retrieval performance on QReCC test set and 3 subsets. DIALKI (mix) beats the supervised T5QR model on all retriever system and test set combinations. * OOD (out-of-domain): only appear in the test set.	61
4.2	Passage retrieval performance of QR models, comparable to scores in Anantha et al. [2021] by using the same BM25 retriever for QReCC test set. DIALKI achieves <i>state-of-the-art</i> results. Recall@10 and Recall@100 are abbreviated as R10 and R100.	63
4.3	Performance of using different retriever inputs for <i>Topic-Concentrated</i> or <i>Topic-Shifted</i> examples.	65
4.4	Average number of tokens (L) and the percentage of overlapping tokens (OL) with the gold passage(s) in output rewrites.	67
4.5	Examples of predicted rewrites and the gold passage ranks by using them as the DE retriever input. <i>The gold answer is italicized in the gold passage.</i>	67

4.6	Examples of predicted rewrites and the gold passage ranks by using them as the BM25 retriever input. <i>The gold answer is italicized in the gold passage.</i>	68
4.7	Error analysis example 1: DIALKI (mix) rewrite contains less context than the human rewrite, which leads to worse ranking of the gold passage.	68
4.8	Error analysis example 2: DIALKI (mix) rewrite contains a misinterpretation of the user request, which leads to worse ranking of the gold passage than the human rewrite.	69
4.9	Results of using the dialogue context, predicted rewrite or human rewrite as the retriever input with the <i>finetuned</i> DE as the retriever.	70
5.1	Overall statistics of INSCIT.	80
5.2	Agent response strategy statistics. DIRECT, CLAR, REL, and NI indicate direct answer, clarification, no direct but relevant answer, and no information.	81
5.3	Examples of clarification and no-direct-but-relevant-answer agent responses. <i>Factual information from evidence passages is italicized in agent responses.</i>	81
5.4	Examples of two agent reference responses with different response strategies.	82
5.5	An example of consecutive clarifications.	82
5.6	Automatic scores on the dev set.	87
5.7	Automatic scores on the test set, and human scores on 50 <i>sampled</i> test conversations (290 turns) for dimensions rated with Likert scales: evidence utility (EU), factual consistency (FC) and coherence (CO).	87
5.8	<i>Passage retrieval</i> results. PT and FT refer to pretraining on TopioCQA and finetuning on INSCIT.	88
5.9	Automatic RG scores for FiD with inputs: dialogue context (DC), oracle evidence passages (OEP), and different (oracle/predicted/no) response types (RT).	90
6.1	Results on the REALTOXICITYPROMPTS test set.	99
6.2	Human pairwise comparison on information completeness (<i>comp.</i>) , where win/lose refers to FINE-GRAINED RLHF.	104
6.3	Automatic evaluation on the QA-FEEDBACK test set.	105

6.4	Automatic evaluation results (test set) of FINE-GRAINED RLHF trained with different reward model weight configurations.	105
6.5	Ablation of reward models on the development set. R_{ϕ_1} , R_{ϕ_2} , and R_{ϕ_3} correspond to the reward model for relevance, factuality, and information completeness.	106
A.1	Qualitative examples on LM customization.	146
A.2	Examples of errors in LM outputs of SFT, Pref. RLHF, and our FINE-GRAINED RLHF. We mark the C_1 (irrelevance, repetition, or incoherence) and C_2 (incorrect or unverifiable facts) types of errors in the model outputs.	148
A.3	Qualitative examples on another question. Here we highlight the information in blue that is covered by the LM trained with FINE-GRAINED RLHF but missed (C_3 : incomplete information error) by SFT and Pref.RLHF.	149
A.4	Qualitative examples on the reward model ablation experiment. We show the LM outputs when we use all the reward models and when one of the reward models is removed during training. We also mark the errors in each output.	151
A.5	Comparison between ChatGPT outputs and FINE-GRAINED RLHF outputs.	153

Chapter 1

Introduction

People seek information on a daily basis. As the accessible information has expanded exponentially with the digital age, developing models to automatically support such information needs becomes increasingly important. Humans often have exploratory information needs which lead them to ask sequential questions, which calls for the development of AI agents that can interact or converse with the human users to provide information. Furthermore, in practice, humans do not always come up with very descriptive queries, in which case an AI agent should converse with them to collaboratively find the information that they are interested in, as a human agent would.

To conduct such an information-seeking conversation with a human user, an AI agent usually interacts with both the user and a knowledge source. Specifically, after the user sends an initial or follow-up information request in natural language, the agent interprets it and searches for relevant information from the knowledge source. Then the agent composes a response to the user that is grounded in the acquired knowledge. Information-seeking dialogues can be grounded in knowledge of different formats like unstructured texts [Choi et al., 2018; Reddy et al., 2019; Feng et al., 2020; Anantha et al., 2021; Adlakha et al., 2022], structured knowledge graphs [Iyyer et al., 2017; Saha et al., 2018] and/or visual knowledge like images [Zhao et al., 2018]. In this thesis, we focus on building agents for information-seeking dialogues grounded in textual documents or passages. We aim to propose solutions to research challenges for such an AI agent in the above interactive information seeking process. In the rest of the introduction, we outline our proposed solutions to three challenges: 1) identifying relevant knowledge from the information source, 2)

collaborating with humans to find information and 3) generating responses with optimized utility to users.

1.1 Knowledge Retrieval and Identification in Conversations

Knowledge identification aims to locate the specific information from grounding documents for response generation. Early benchmarks of document-grounded information-seeking dialogues [Choi et al., 2018; Reddy et al., 2019] mostly consider a single short grounding passage. Since then, an increasing number of studies [Feng et al., 2020; Adlakha et al., 2022; Anantha et al., 2021] introduce new datasets with much longer grounding documents. They also propose models for knowledge identification, and these models are mostly borrowed from existing question answering models [Devlin et al., 2019; Yang et al., 2019; Karpukhin et al., 2020] without dialogue structure modeling. In this thesis, we propose DIALKI [Wu et al., 2021d] for the problem of knowledge identification for response generation in document-grounded information-seeking dialogues that leverage conversation features (e.g., dialogue structure) as well as document structures. We show the effectiveness of DIALKI in knowledge identification on two public benchmarks [Feng et al., 2020; Dinan et al., 2019] for which the identified knowledge by DIALKI greatly improves model performance in response generation.

In practice, the knowledge source for information seeking can often be enormous and contains millions or more passages. Passage retrieval [Voorhees and Tice, 2000; Bajaj et al., 2018] is the task that aims to identifying a small set of passages from a large corpus that contain relevant information to a given user query. Compared to standard retrieval tasks that only handle standalone information queries, *conversational passage retrieval* poses unique challenges in understanding the current user question, as each question needs to be interpreted within the dialogue context. On the other hand, it is not always practical to retrain well-established retrievers (e.g., a search engine) to handle conversational queries specifically. In this case, the conversational agent should learn to effectively query a given retriever which can only be seen as a black-box. The task of query rewriting (QR) [Elgohary et al., 2019; Dalton et al., 2020] in a conversation is to convert a context-dependent question into a self-contained question, which enables the use of any off-the-shelf retriever. Previous work that leverages QR for conversational passage retrieval [Anantha et al., 2021; Dalton et al., 2020] collects human-written rewrites to train a supervised QR model. However, there exist certain limitations of such QR models. First, they rely on human rewrites for supervised training. In

addition, when collecting such rewrites, humans are usually instructed to rewrite conversational queries to be unambiguous to a human outside the dialogue context. This can make human rewrites sub-optimal for the retrieval performance. In addition, the resulting QR model is agnostic to the downstream retriever. To address these issues, we propose CONQRR [Wu et al., 2022] that trains a QR model to directly optimize for retrieval performance. CONQRR adopts a reinforcement learning (RL) to train a QR model, optimizing a reward coming from the retrieval performance by querying the black-box retriever with the re-written query. An advantage of this approach is the training does not have to rely on human rewrites, i.e., human annotation costs are reduced.

1.2 Interactive and Collaborative Information Seeking

One important advantage of conversational information-seeking systems is that users do not need to come up with a very descriptive query by themselves. In realistic settings, users can start with a request that is under-specified, and through conversational interactions, the agent can *collaboratively* guide users to refine or clarify their intent. In addition, when users start with a request that has no direct answer, the agent can *collaboratively* relax their queries and proactively suggest relevant information that may partially satisfy the user’s information needs. Most existing information-seeking dialogue datasets have a very restricted format (e.g., consecutive question and answer pairs) [Choi et al., 2018; Reddy et al., 2019; Anantha et al., 2021; Adlakha et al., 2022]. There exists work that studies clarification questions but they use artificially created data [Feng et al., 2020; Saeidi et al., 2018; Aliannejadi et al., 2021], which fails to capture natural information-seeking interactions.

To address the above issue, we build a new dataset INSCIT containing information-seeking dialogues that cover a variety of conversation topics and are collected from human-human conversations with natural interactions like clarification [Wu et al., 2023b]. We follow the dialogue data collection pipeline in Wen et al. [2017] to collect human-human dialogues with careful quality control mechanisms. We propose two subtasks for this dataset—passage identification and response generation, as well as a novel paradigm for human evaluation that is missing in most work of document-grounded information-seeking dialogues. Finally, we propose a model that leverages DIALKI [Wu et al., 2021d] to address the challenge of limited training data for both subtasks in INSCIT.

1.3 Response Generation Informed by Human-centered Objectives

As the essential goal of an AI agent for information-seeking users is to provide helpful information, a natural question one may ask is what kind of agent response the user would find to be the most helpful? Intuitively, to teach the AI agent to optimize the response helpfulness, we can have users act as evaluators and provide feedback. This allows us to directly train the model with *human definitions* of helpfulness.

Reinforcement learning from human feedback (RLHF) [Ziegler et al., 2019] is an increasingly popular training method to optimize model generations towards human objectives, due to its success in building state-of-the-art AI models [Ouyang et al., 2022; OpenAI, 2023; Fu et al., 2023]. In a standard RLHF pipeline, the first step is to collect preference feedback. Humans are presented with two or more outputs and asked to select one or rank them. This signal is then used to train a reward model, which computes a single scalar reward for each LM output. Then, in the second step, the LM is trained with a reinforcement learning algorithm called proximal policy optimization (PPO) to optimize the reward it receives from the reward model. However, such human preferences over model outputs provide very coarse-grained feedback and do not necessarily capture clear human-centered objectives. Intuitively, fine-grained feedback can be more explicit and reliable. Therefore, we propose to localize and categorize human feedback, to get explicit information about which part of the model output has what type of problem. To leverage that feedback, we propose a language model training framework called FINE-GRAINED RLHF [Wu et al., 2023a] to train a language model to optimize such specific human objectives.

1.4 Thesis Overview

To summarize, this thesis addresses three challenges towards building an interactive AI agent to support the needs of information seeking users. The remainder of this thesis is organized as follows. In Chapter 2, we provide detailed background and related work for interactive information seeking, with a focus on the challenges described above that this thesis targets. In Chapter 3-4, we present the models we propose for knowledge identification and retrieval in document-grounded information-seeking dialogues, namely DIALKI and CONQRR. In Chapter 5, we introduce INSCIT, the dataset we collect to facilitate research in interactive and collaborative information-seeking, as well as the proposed tasks, evaluation and models

along with the dataset. Chapter 6 presents FINE-GRAINED RLHF that trains a model to generate responses informed by specific human feedback. We conclude in Chapter 7 by summarizing the work, and identifying some future directions for building AI agent that more effectively support interactive information seeking.

Chapter 2

Background

Before introducing our proposed solutions to research challenges explained in Chapter 1, in this Chapter, we first provide a background work review of interactive AI agents for information-seeking. Upon receiving a user’s request in a conversation, an AI agent for information-seeking tasks should first identify relevant information from a knowledge source and then compose a response grounded in this information. In § 2.1, we first briefly review existing work that aims to build conversational models that ground their response generation in external knowledge. We categorize and differentiate these *grounded* dialogue systems by their target conversation types including information-seeking conversations. In § 2.2, we provide a deeper dive in related work of information-seeking conversations grounded in documents specifically, which is the focus of this thesis. Finally, in § 2.3, we provides an overview of existing language model training approaches involving reinforcement learning and learning from human feedback, which closely relate to our proposed methods CONQRR and FINE-GRAINED RLHF that will be introduced in Chapter 4 and Chapter 6.

2.1 Grounded Dialogue Systems

Grounded dialogue systems aim to condition their response generation on the given external information like textual knowledge sources or a visual setting. This concept has been applied to a variety of dialogues including social chats, task-oriented and info-seeking dialogues. In general, grounding in the given information leads to a conversation that is more relevant and consistent with that information. In addition, unlike ungrounded conversational models, a grounded model will be able to conduct conversations about informa-

tion that is unseen during training. In this section, we categorize existing work in dialogue systems grounded in textual knowledge sources by their target conversation types and introduce their associated task or goals.

2.1.1 Social Chats

The main goals of social chats are to keep users engaged and to deliver information. Most existing work incorporates external knowledge into social chat systems in order to make the conversations more engaging and informative. Conversations for social chats are often very casual and users are not expected to have a clear information need, which is different from information-seeking conversations. Ghazvininejad et al. [2018] develop a memory network based model to conduct casual conversations in the restaurant domain by leveraging tips from Foursquare. Dinan et al. [2019] crowd-source conversations where each utterance is grounded in up to one single sentence from 7 Wikipedia paragraphs. Zhou et al. [2018b] and Moghe et al. [2018] collect movie discussion conversational datasets grounded on movie plots and infoboxes from sources like Wikipedia and IMDB. Zhou et al. [2018a] leverage ConceptNet as a commonsense knowledge graph to enhance response generation.

2.1.2 Information-Seeking Dialogues

Different from social chat, information-seeking dialogues, the focus of this thesis, are generally conducted when the user has one or a sequence of specific information requests. In response, the agent is targeted to find and provide such requested information to the user. Here, we outline previous work under different task settings for information-seeking conversations. We further discuss existing modeling and evaluation work in § 2.2.

The task of conversational question answering (CQA) or reading comprehension [Choi et al., 2018; Reddy et al., 2019] falls into this category and is generally defined as the task of answering a series of interconnected questions about a short text passage in a conversation. More recently, some work also propose to have other types of grounding knowledge sources including knowledge graphs [Iyyer et al., 2017] or a corpus with millions of documents in the open-domain setting [Anantha et al., 2021; Adlakha et al., 2022]. Most CQA datasets have the strict format of question and answer pairs in each conversation and some of them restrict each answer to be a short string span in the grounding passage.

In contrast, a few studies explore more diverse scenarios in information-seeking dialogues like asking for clarification. Saeidi et al. [2018] introduces a conversational machine reading dataset to address under-specified questions in the regulatory domain (e.g., government policies) by requiring agents to ask follow-up questions grounded in a short text snippet that are answerable with “yes / no” answers. Aliannejadi et al. [2021] collect data for asking clarification question(s) to an ambiguous or faceted user query in order to provide a specific answer, with up to 3 user-agent turns in a conversation. Their conversations are collected based on synthetic rules or made-up dialogues based on predefined dialogue scenes, instead of being naturally collected as human-human dialogues. Feng et al. [2020] and Gustavo Penha and Hauff [2019] provide conversations with varied user intents that are grounded in documents from social welfare websites or the Stack Exchange website. Feng et al. [2020] also have their conversations collected in a rather artificial way as they first synthesize the flow of grounding knowledge in a conversation and then asks an annotator to write utterances based on the pre-determined grounding knowledge. Gustavo Penha and Hauff [2019] heuristically define hyperlinks included in conversations from the Stack Exchange website as the grounding document and most of their conversations only consist of 2 user-agent turns. The lack of datasets and benchmarks for information-seeking conversations with diverse and collaborative interactions motivates us to collect INSCIT [Wu et al., 2023b] with over 800 human-human conversations, introduced in Chapter 4.

2.1.3 Task-Oriented Dialogues

There are also other knowledge-grounded dialogues that involve assisting the user with a task, such as restaurant reservations or flight ticketing booking. Task-oriented dialogues [Williams et al., 2013; Wen et al., 2017; Budzianowski et al., 2018] rely on structured data, with a task specific schema as well as a database, which can be the same for all dialogues performing the same task. For example, for the task of restaurant reservations, the schema can consist of required slots like “location”, “price range”, “date” and so on. The corresponding database usually stores all available restaurants for reservation with information related to the slots mentioned above. Such schema and databases are considered as the grounding environment for task-oriented dialogues, although the way the conversational agents normally retrieve information from the environment is different from agents that perform social chats or provide information for user requests as discussed in the above two sections. Task-oriented dialogue systems normally track and update the values for

slots in the schema throughout the conversation and form a SQL query to retrieve results from the database at the end of each user turn. Success is evaluated based on retrieved results.

Another type of task-oriented system is a conversational recommender system [Li et al., 2018; Jannach et al., 2021], which is designed to allow the system to provide explanation for the recommended items (e.g., movie, product or information), refine their recommendation based on user feedback, or even help the user discover their preferences through conversational interactions. These systems usually target to optimize the user satisfaction with recommended items. Such conversations are grounded in information about items available for recommendation, which can be structured [Liu et al., 2020; Volokhin et al., 2021] or unstructured [Kang et al., 2019]. They can also ground in user information like past interactions [Thompson et al., 2004].

2.2 Document-Grounded Information-Seeking Dialogues

Among all the different types of knowledge grounded dialogue systems, this thesis will specifically focus on addressing some key issues in *information seeking dialogues* grounded in *documents*. We discuss more details about the evaluation metrics and existing models below.

2.2.1 Evaluation

A few early CQA benchmarks [Choi et al., 2018; Reddy et al., 2019] only consider extractive agent responses (i.e., a string span from the given grounding passage). In other words, the response generation task in such dialogues is essentially a knowledge span extraction problem. Therefore, they simply adopt question answering evaluation metrics [Rajpurkar et al., 2016] including exact match and word-level F1 scores.

In contrast, more recent benchmarks have abstractive agent responses in conversations, that are grounded in a much longer document or a large corpus of documents [Anantha et al., 2021; Adlakha et al., 2022; Feng et al., 2020]. Therefore, most of these benchmarks consider separate evaluations for response generation and knowledge identification. We discuss the current commonly used metrics for evaluating response generation and knowledge identification respectively below.

Response Generation Most commonly used automatic evaluation metrics for response generation in an information-seeking dialogue are untrained automatic metrics like BLEU [Papineni et al., 2002] and word-level F1-score. These metrics are simply calculated based on string overlap between the generated response and one or a set of reference responses. SacreBLEU Post [2018] is the most commonly used implementation for BLEU to reduce inconsistencies in the score reporting due to different implementation variations. Recently, some work [Sellam et al., 2020; Hashimoto et al., 2019] proposes model-based evaluators to be trained on human judgment data to mimic human judges on generated text quality like naturalness. However, due to their complexity and potential problems such as overfitting and “gaming of the metric” [Gao et al., 2018; Celikyilmaz et al., 2020], these model-based evaluators have not been widely used. As a result, We follow most previous work to use SacreBLEU and token-level F1-score to automatically evaluate response generation in Chapter 3 and Chapter 5.

Although automatic evaluation is more efficient and useful for model development, human evaluation is usually seen as more important for response generation as automatic metrics can be unreliable and inconsistent with human decisions [Mathur et al., 2020; Celikyilmaz et al., 2020]. However, most information-seeking dialogue studies mentioned above only report automatic evaluation on response generation. We only found one follow-up study [Gao et al., 2022] that includes human evaluation in their results. They give human evaluators the dialogue context and the generated response to give scores on dimensions like “relevance” and “informativeness.” Such evaluation overlooks the importance of evaluating the consistency between the grounding knowledge and generated responses. Although human evaluation on response generation in those document-grounded information-seeking dialogues has not yet brought much attention, it has been widely used in evaluating knowledge-grounded social chats. Most existing work [Ghazvininejad et al., 2018; Zhou et al., 2018b; Moghe et al., 2018; Dinan et al., 2019; Qin et al., 2019] present the dialogue context to human raters and ask them to rate or compare generated responses from different systems in aspects like “fluency,” “appropriateness” and “informativeness,” or simply with an overall score. A few present grounding knowledge to the user raters to evaluate the consistency between the response and grounding knowledge [Wu et al., 2021c; Shuster et al., 2021]. We also conduct human evaluation in Chapter 5.

Knowledge Identification Knowledge identification (KI) is the task of locating knowledge in the given grounding knowledge source (e.g., a long or multiple documents) that is relevant to the current user query

given the conversation context. Depending on the task setup, the knowledge to be identified can be one or a set of string spans, sentences or paragraphs. Feng et al. [2020] define the knowledge used in each agent turn to be a string span and as a result, they adopt automatic evaluation metrics including exact match and word-level F1 scores as used for question answering tasks [Rajpurkar et al., 2016]. Anantha et al. [2021] and Adlakha et al. [2022] introduce datasets for open-domain CQA which requires the agent to perform knowledge passage retrieval first before generating the response. Therefore, KI in their settings is basically a passage retrieval problem and is evaluated with retrieval metrics like Recall at K and mean reciprocal rank (MRR). Following such previous work, we report exact match and word-level F1 scores in Chapter 3 for knowledge identification within a document, and Recall at K and MRR in Chapter 4 for passage retrieval.

2.2.2 Models

Many existing models [Feng et al., 2020; Anantha et al., 2021; Adlakha et al., 2022] are largely borrowed from models for question answering tasks [Devlin et al., 2019; Yang et al., 2019; Karpukhin et al., 2020], without explicit modeling of dialogue structure. Intuitively, models for other knowledge-grounded dialogues like social chats should also be able to be adapted to these information-seeking dialogues as long as they have the same grounding knowledge format (e.g., documents). However, most such prior work has explored architectures for document-grounded dialogues in an end-to-end framework, optimizing a loss function targeting response generation only [Ghazvininejad et al., 2018; Zhou et al., 2018b; Yavuz et al., 2019; Zhao et al., 2020; Lin et al., 2020b; Li et al., 2019b]. Only a few works [Dinan et al., 2019; Lian et al., 2019; Kim et al., 2020; Zheng et al., 2020] build models for and evaluate on the knowledge identification task. Specifically, their models aim to identify a string span or a sentence from a given document. These models separately encode sentences in documents, which may have strong contextual dependencies among each other. As described in Chapter 3, our work [Wu et al., 2021d] proposes a KI model DIALKI that leverages document structure dividing each document into multiple passages to process, and applies a multi-task learning framework to model relations between grounding documents and history turns. We then show that better knowledge identified by DIALKI leads to better response generation performance. Similar to DIALKI, Kim et al. [2020] and Zheng et al. [2020] incorporate previously used knowledge, but they use a single vector to sequentially track the state of the used knowledge.

In an open-domain scenario where a large document or passage corpus instead of a single document is given as the knowledge source, a conversational agent needs to retrieve the most relevant passage(s) before generating an answer to the question. Most existing conversational retrieval models require fine-tuning a retriever of a specific type. Yu et al. [2021], Lin et al. [2021] and Kim and Kim [2022] attempt to fine-tune a dual encoder retriever [Xiong et al., 2021; Karpukhin et al., 2020] to handle conversational queries. Kumar and Callan [2020] propose a framework focusing on improving the passage re-ranker after the retrieval. However, it is not always practical to fine-tune retrievers for handling conversational user queries, as discussed in Chapter 1. In order to directly use an *off-the-shelf* retriever, conversational *query rewriting* (QR) [Elgohary et al., 2019] has been applied in prior work [Vakulenko et al., 2021; Lin et al., 2020a; Yu et al., 2020; Voskarides et al., 2020] to first convert a conversational query into a standalone one. Yu et al. [2020] propose a supervised QR model trained with human rewrites and weak QR supervisions specifically for conversational search tasks that are generated from additional search session resources. Lin et al. [2020a] and Vakulenko et al. [2021] also use human rewrites to train a supervised QR model based on pre-trained language models like T5 [Raffel et al., 2020] or GPT2 [Radford et al., 2019]. Voskarides et al. [2020] use human rewrites to train a model that classifies whether each token in the dialogue context should be used to construct the query for retrieval. In contrast, in Chapter 4, we show the limitations of human rewrites used as QR supervision and design reinforcement learning based QR model which can achieve better performance than supervised models even without human rewrites.

2.3 Learning from Human Feedback

2.3.1 Reinforcement Learning (RL) for NLP

Prior work applies RL approaches to address text generation tasks like machine translation [Ranzato et al., 2016; Wu et al., 2016], text summarization [Paulus et al., 2018; Celikyilmaz et al., 2018] and image captioning [Rennie et al., 2017; Fisch et al., 2020] by training a model directly optimized towards generation quality metrics like BLEU, ROUGE or CIDEr. Most of these works use a vanilla policy gradient or a self-critical sequence RL training algorithm [Rennie et al., 2017]. Ouyang et al. [2022] use reinforcement learning to improve the instruction-following capability of large language models, where the reward comes from a

trained reward model from collected human feedback. Specifically, they use an actor-critic RL algorithm called Proximal Policy Optimization (PPO) [Schulman et al., 2017] for training. This line of work closely relates to our proposed framework FINE-GRAINED RLHF in Chapter 6, and we provide more details below.

2.3.2 Reinforcement Learning from Human Feedback (RLHF)

RLHF [Ziegler et al., 2019; Xu et al., 2023a; Ouyang et al., 2022] aims to optimize the policy language model to generate content that is desired by the human user. This framework has been explored to improve the model performance on a variety of natural language processing tasks such as text summarization [Stiennon et al., 2020], instruction following [Ouyang et al., 2022], question answering [Menick et al., 2022; Nakano et al., 2021] and reducing harmfulness [Bai et al., 2022b,a; Lu et al., 2022; Ganguli et al., 2023]. Most of these studies collect human preferences over pairs of model outputs on one or a set of desired attributes, which is used to train a reward model to assign a holistic score for a generation output during RL training. Glaese et al. [2022] trains separate reward models that assign scores for different desired attributes, but still uses a single reward that combines scores from all reward models. In contrast, our FINE-GRAINED RLHF [Wu et al., 2023a] introduced in Chapter 6 explores RLHF with fine-grained reward models trained on human feedback where each reward model provides a reward after every small text segment for a specific type of desired behavior.

2.3.3 Non-RL Learning from Human Feedback in NLP

There also exists work that explores non-RL methods to learn from human feedback. Yuan et al. [2023] trains a reward model that predicts a single holistic score for each model output and selects samples with the highest reward scores for supervised fine-tuning, which is often called rejection sampling. Touvron et al. [2023] show the effectiveness of RLHF compared to rejection sampling. Shi et al. [2022]; Hancock et al. [2019]; Xu et al. [2023a] train a conversational model to predict both the response and a binary user satisfaction score in order to improve the response generation. Besides numerical or preference human feedback that is often used in RLHF, natural language (NL) human feedback has also been explored. Madaan et al. [2022]; Dalvi Mishra et al. [2022] collect and store NL human feedback in a feedback memory for the model to retrieve and then perform the end task conditioning on the retrieved feedback. Chen et al. [2023];

Scheurer et al. [2023, 2022] use a refinement model to refine model outputs conditioning on NL human feedback and then use a reward model to select the best refined outputs for supervised fine-tuning. Paul et al. [2023] explore using model-generated NL feedback to iteratively improve LM performance on reasoning tasks. Methods for using a reward model to guide LM generation towards desired behaviors at inference time [Liu et al., 2021; Dathathri et al., 2020] can complement our work that aims to improve the LM during training. Korbak et al. [2023] also explore incorporating human feedback into LM pre-training by using strategies like conditional training. Specifically, they append control tokens associated with preferences to each textual segment in the training data, labeled by a reward model trained from human feedback.

Chapter 3

DIALKI: Knowledge Identification in Conversational Systems through Dialogue-Document Contextualization

3.1 Introduction

Many conversational agent scenarios require knowledge-grounded response generation, where knowledge is represented in written documents. Most prior work has explored architectures for knowledge grounding in an end-to-end framework, optimizing a loss function targeting response generation [Ghazvininejad et al., 2018; Zhou et al., 2018b; Yavuz et al., 2019]. However, the knowledge needed at any one turn in the dialogue is typically localized in the document, and some studies have shown that directly optimizing for knowledge extraction helps resolve complex user queries [Feng et al., 2020] and increases user

Dialogue Context

[User]: Hi, can you tell me something about the private service bureau licenses?

[Agent]: Do you want to apply for a PSB?

[User]: No, I was being curious. Just in case, what should I do if I apply for PSB?

[Agent]: Your application will be reviewed in Albany's DMV. After that, it will be sent to your local DMV office and you'll be scheduled for an inspection.

Grounding document

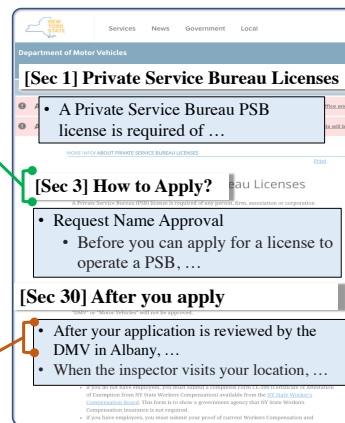


Figure 3.1: In a document-grounded conversation, *knowledge identification* targets to locate a knowledge string within a long document to assist the agent in addressing the current user query.

engagement in social chat [Dinan et al., 2019; Moghe et al., 2018]. For long documents, explicit knowledge identification can also be useful for model interpretability and human-in-the-loop assistant scenarios.

Following Feng et al. [2020], we define *knowledge identification* as the task of locating knowledge in a long document that is relevant to the current user query given the conversation context (Figure 3.1). Knowledge identification is similar to open question answering [Chen et al., 2017; Min et al., 2019], the task of answering a factoid question given a large grounding, except that it is not an interactive setting like dialogues. With the assumption of a long grounding document, our task differs from prior work in conversational question answering [Choi et al., 2018; Reddy et al., 2019], which focuses on answering a sequence of factoid questions about a short text snippet. Additionally, real user information needs can involve conversations with diverse forms of user queries and dialogue acts (e.g., asking for user preference, etc) as shown in Figure 3.1. Previous work in knowledge identification encodes the grounding document as a single string [Feng et al., 2020], or splitting it into isolated sentences [Dinan et al., 2019; Kim et al., 2020; Zheng et al., 2020] which potentially loses important discourse context.

In this chapter, we introduce DIALKI to address knowledge identification in conversational systems with long grounding documents. In contrast to previous work, DIALKI extends multi-passage reader models in open question answering [Karpukhin et al., 2020; Cheng et al., 2020] to obtain dense encodings of different spans in multiple passages in the grounding document, and it contextualizes them with the dialogue history. Specifically, DIALKI extracts knowledge given a long document by dividing it into paragraphs or sections and individually contextualizes them with dialogue context. It then extracts knowledge by first selecting the most relevant passage to the dialogue context and then selecting the final knowledge string within the selected passage. Processing each passage rather than the full document greatly shortens the knowledge context, while preserving enough discourse context for reasoning. DIALKI also uses a multi-task objective to identify knowledge for the next turn, as well as used knowledge for previous turns that helps improve the learning of both dialogue and document representations by capturing interdependencies between the next agent utterance, previous utterances, and the grounding document.

Our model significantly improves over baselines on two conversational datasets, particularly on previously unseen documents or topics. Analyses show good generalization to longer grounding documents and longer dialogue context, as well as improvements in response generation. Model robustness can be further

improved with an f-divergence based posterior regularization.

This chapter contains material that was originally published in Wu et al. [2021d]. Bo-Ru Lu and I were the main contributors of this work. We discussed and wrote the paper with the other collaborators throughout the project. I initiated the project idea and made the first implementation of the system. Bo-Ru Lu refactored the codebase for more efficient experimentation. We made equal contributions in conducting experiments and analysis.

3.2 Method

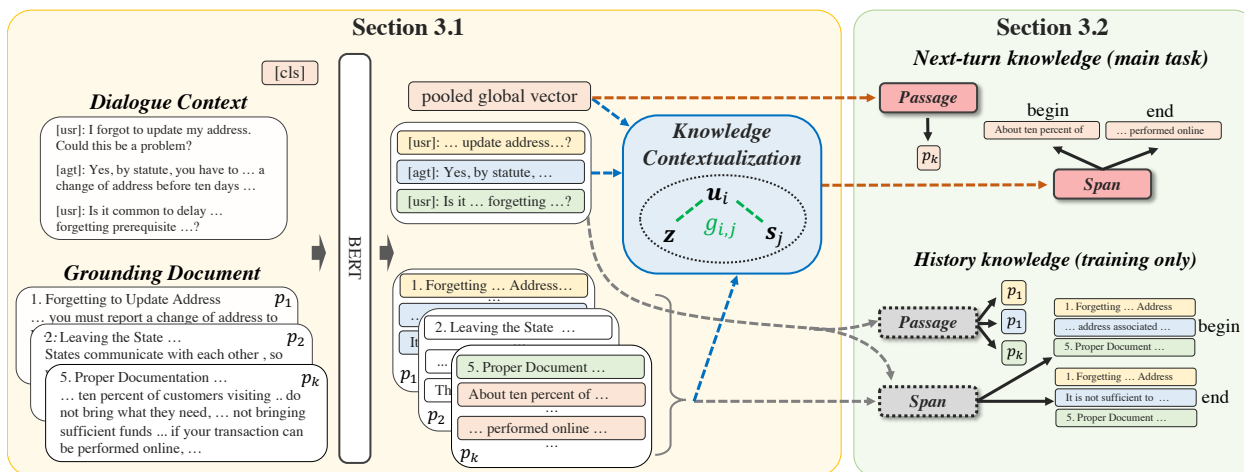


Figure 3.2: The overview of DIALKI. Each document is divided into passages. We apply BERT and a knowledge contextualization mechanism to obtain dialogue context and knowledge representations (left), for performing both next (main) and history (auxiliary) turn knowledge identification tasks (right). For each turn, DIALKI identifies knowledge by selecting the relevant passage as well as the begin/end spans in the passage.

Problem Definition *Knowledge identification* in a document-grounded dialogue is defined as follows. At inference, given the dialogue context consisting of a sequence of n previous utterances (u_1, u_2, \dots, u_n) and a grounding document \mathcal{D} , select a substring y in \mathcal{D} that is relevant to the dialogue context and will be used in the next turn (i.e., utterance) in a conversation. We denote u_1 as the last (user) turn and u_n as the earliest turn in the available dialogue history, which may be truncated.

Each document consists of a sequence of passages $\mathcal{D} = \{p_1, p_2, \dots, p_{|\mathcal{D}|}\}$ based on paragraphs or sections. Each passage p consists of a sequence of semantic units $p = (s_1, s_2, \dots, s_l)$, where each semantic

unit (SU) can be either a token or a span or a sentence depending on how the document is segmented. For simplicity, we use “span” as the semantic unit (SU) in this section to describe our model.

Method overview In this section, we introduce DIALKI, a multi-task learning model for knowledge identification as illustrated in Figure 3.2. We first introduce how we obtain dialogue utterance and knowledge span representations from BERT [Devlin et al., 2019] and a span-level knowledge contextualization mechanism (§ 3.2.1). These representations are then used for knowledge identification in our multi-task learning framework, which includes the main task of next-turn knowledge identification and an auxiliary task of history knowledge prediction applied during training only (§ 3.2.2). Finally, we describe our joint training objective and inference details (§ 3.2.3).

3.2.1 Encoding Dialog Context and Knowledge

Multi-Passage Encoding

Here, we describe how we initially obtain dense vector representations of each passage in the grounding document as a set of span representations. Inspired by the recent open-domain question answering multi-passage reader models [Karpukhin et al., 2020; Cheng et al., 2020], we concatenate the dialogue context (u_1, u_2, \dots, u_n) with u_1 to be the most recent user turn, the document title t and each passage p , and use a pretrained language model like BERT [Devlin et al., 2019] to encode the concatenated sequence. More formally, the model input \mathbf{X} for a passage p of length l becomes:

$$\mathbf{X} = [\text{cls}][\text{usr}] u_1 [\text{agt}] u_2 \cdots [\text{usr}] u_n \\ [\text{sep}] t [\text{cls}] s_1 [\text{cls}] s_2 \cdots [\text{cls}] s_l [\text{sep}]$$

where ‘[usr]’ and ‘[agt]’ are special tokens indicating the start of a user or agent turn. ‘[cls]’ indicates the start of the whole sequence or each knowledge span. ‘[sep]’ are separator tokens. Then we encode \mathbf{X} and gather a sequence of pooled output vectors $\mathbf{H} = G(\text{BERT}(\mathbf{X}))$ where $G(\cdot)$ gathers vectors of all ‘[cls]’, ‘[usr]’ and ‘[agt]’ tokens. We decompose \mathbf{H} as $[\mathbf{z}, \mathbf{u}_1, \dots, \mathbf{u}_n, \mathbf{s}_1, \dots, \mathbf{s}_l]$ where \mathbf{z} , \mathbf{u}_i , \mathbf{s}_j are pooled representations of the whole input (the first ‘[cls]’ token in \mathbf{X}), dialogue utterance u_i and span s_j respectively.

Knowledge Contextualization

We leverage the pooled global, utterance and span representations $\mathbf{z}, \mathbf{u}_i, \mathbf{s}_j$ obtained above for each passage p to further contextualize knowledge span representations. Inspired by how EntNet [Henaff et al., 2017] updates each entity representation based on an input sequence, for each span s_j we calculate an updated span embedding $\hat{\mathbf{s}}_j$ contextualized with the sequence of previous *user* utterance representations as below. We use a gating function g which determines how much the span embedding should be updated based on \mathbf{u}_i and \mathbf{z} .

$$\begin{aligned} \mathbf{a}_{i,j} &= \mathbf{W}_s \mathbf{s}_j + \mathbf{W}_z \mathbf{z} + \mathbf{W}_u \mathbf{u}_i, \quad i \in \mathbf{C}_u \\ \hat{\mathbf{s}}_j &= v \left(\sum_{i \in \mathbf{C}_u} \left[\phi(\mathbf{a}_{i,j}) \odot g_{i,j} \right] + \mathbf{s}_j \right) \\ g_{i,j} &= \sigma \left(\mathbf{u}_i^\top \mathbf{z} + \mathbf{u}_i^\top \mathbf{s}_j \right), \end{aligned}$$

where $\mathbf{W}_s, \mathbf{W}_z, \mathbf{W}_u \in \mathbb{R}^{d \times d}$ are model parameters, and \mathbf{C}_u indexes the most recent user turns. $v(\cdot)$ is the vector normalization operation. σ is the sigmoid function and ϕ can be any non-linear activation function. In our model, we use ReLU for ϕ . Similarly, we calculate $\tilde{\mathbf{s}}_j$ with previous *agent* turns. Then new span embedding denotes as $\hat{\mathbf{s}}_j = [\mathbf{s}_j, \hat{\mathbf{s}}_j, \tilde{\mathbf{s}}_j]$. In our experiments, we set \mathbf{C}_u to contain the most recent 2 user turns only, which leads to the best results.

3.2.2 DIALKI with Multi-Task Learning

Next-Turn Knowledge Identification

The main task in DIALKI is next-turn knowledge identification consisting of two parts: passage prediction and knowledge span prediction (upper right in Figure 3.2). At inference, given a set of passages $\{p_1, p_2, \dots, p_{|\mathcal{D}|}\}$ and the dialogue context (u_1, u_2, \dots, u_n) , our multi-passage knowledge identification model targets to predict the most relevant passage p_k based on the softmax probability in Eq. (3.1), as well as the most relevant knowledge string $y = (s_b \dots s_e)$ in p_k to the next agent turn in the conversation, based on the begin and end span probabilities in Eq. (3.2-3.3)

We obtain the pooled global vector \mathbf{z} , each utterance and span representation $\mathbf{u}_i, \hat{\mathbf{s}}_j$ for each passage p

as described in § 3.2.1. We denote \mathbf{Z} as the matrix containing the pooled global vectors for all passages and \mathbf{U}_i as utterance representations for u_i in all passages. $\hat{\mathbf{S}}$ denote the matrix with all span representations after knowledge contextualization.

Training Objectives Eq. (3.1-3.3) show objective functions of knowledge passage \mathcal{L}_{psg} , begin $\mathcal{L}_{\text{begin}}$ and end \mathcal{L}_{end} span predictions. $q(\cdot)_t$ denotes the t -th index of the vector resulting from the softmax function. The variables \hat{k} , \hat{b} and \hat{e} correspond to the gold passage, begin and end span indices.

$$\mathcal{L}_{\text{psg}} = -\log q(\mathbf{W}_p \mathbf{Z})_{\hat{k}} \quad (3.1)$$

$$\mathcal{L}_{\text{begin}} = -\log q(\mathbf{W}_b \hat{\mathbf{S}})_{\hat{b}} \quad (3.2)$$

$$\mathcal{L}_{\text{end}} = -\log q(\mathbf{W}_e \hat{\mathbf{S}})_{\hat{e}} \quad (3.3)$$

where $\mathbf{W}_p, \mathbf{W}_b, \mathbf{W}_e \in \mathbb{R}^d$. Therefore, the combined next turn knowledge selection loss function becomes:

$$\mathcal{L}_{\text{next}} = \mathcal{L}_{\text{psg}} + \mathcal{L}_{\text{begin}} + \mathcal{L}_{\text{end}}.$$

History Knowledge Identification

The auxiliary task during training is to predict previously used knowledge, with the intuition that used knowledge in documents by history turns would guide the search for the next knowledge to use.

Training Objectives With the same representations used in next-turn knowledge identification, similar to Eq. (3.1-3.3), we calculate both passage- and span-level prediction losses for each history turn u_i if the knowledge string for u_i can be found in \mathcal{D} . We calculate the passage prediction loss $\mathcal{L}_{\text{psg}}^h$ of previous turns as follows:

$$\mathcal{L}_{\text{psg}}^h = \frac{1}{\|\mathbf{U}^*\|} \sum_{u_i \in \mathbf{U}^*} -\log q(\mathbf{W}_p^h \phi(\mathbf{W}^h \mathbf{U}_i))_{k_i}$$

where $\mathbf{W}^h \in \mathbb{R}^{d \times d}$, $\mathbf{W}_p^h \in \mathbb{R}^d$ are model parameters. \mathbf{U}^* is the set of history turns that can find their knowledge strings in the document \mathcal{D} . k_i is the gold passage index for turn u_i . ϕ is a non-linear activation function, with ReLU used in our model.

Similarly, we calculate the average losses of predicting the gold begin $\mathcal{L}_{\text{begin}}^h$ and end $\mathcal{L}_{\text{end}}^h$ knowledge spans of all history turns in \mathbf{U}^* in their gold passages. To do so, for each u_i , we compute the dot product between a linearly transformed \mathbf{u}_i and each span embedding \mathbf{s}_j in p_{k_i} and apply a cross-entropy loss for the begin or end span prediction. Therefore, the combined knowledge selection loss function of history turns becomes:

$$\mathcal{L}_{\text{hist}} = \mathcal{L}_{\text{psg}}^h + \mathcal{L}_{\text{begin}}^h + \mathcal{L}_{\text{end}}^h.$$

3.2.3 Training and Inference

Posterior Regularization During training, we incorporate a posterior regularization mechanism [Cheng et al., 2021] to enhance the model’s robustness to domain shift. Specifically, we add an additional adversarial training loss as below. Div is some f-divergence.¹ Let x be the encoded \mathbf{X} (defined in § 3.2.1) after the BERT word embedding layer, DIALKI outputs $f_{\text{psg}}(x)$, $f_{\text{begin}}(x)$ and $f_{\text{end}}(x)$ as the next turn passage, begin and end knowledge span logits (the results before the softmax function $q(\cdot)$ in Eq. (3.1-3.3)) respectively.

$$\mathcal{L}_{\text{adv}} = \max_{\|\epsilon\| \leq a} \sum_{f \in \{f_{\text{psg}}, f_{\text{begin}}, f_{\text{end}}\}} \text{Div}(f(x) \| f(x + \epsilon))$$

The above loss function essentially regularizes the g -based worst-case posterior difference between the clean and noisy input (with norm of the added noise no larger than some scalar a) using an inner loop to search for the most adversarial direction.

Joint Training Objective Combining all the above components, our final model optimizes the joint objective \mathcal{L} :

$$\mathcal{L} = \mathcal{L}_{\text{next}} + \alpha \mathcal{L}_{\text{hist}} + \beta \mathcal{L}_{\text{adv}} \tag{3.4}$$

where α and β are tunable hyperparameters.

Inference During inference, we perform next-turn knowledge prediction only. We first predict the passage with the highest probability and enumerate over all possible knowledge span sequences in the selected

¹We use Jensen-Shannon divergence in all experiments.

passage with a maximum length. Then we select the span sequence with the highest score to be the final knowledge string. The score of each span sequence is calculated as the sum of begin and end span probabilities.

3.3 Experiment

3.3.1 Datasets and Evaluation Metrics

Datasets We use two datasets for our experiments: *Doc2Dial* [Feng et al., 2020] and *Wizard of Wikipedia (WoW)* [Dinan et al., 2019]. We follow all original preprocessing and evaluation scripts for data processing of the following datasets.² These original data preprocessing scripts contain the step of downloading data. Doc2Dial Feng et al. [2020] contains about 4.8k English goal-oriented dialogues in 4 social-welfare domains, with an additional Covid-19 domain in the blind held-out test set. Each dialogue has an average of 14 turns, grounded on a long document with more than 1k tokens on average. Each user or agent turn is grounded in a sequence of knowledge spans as annotated in the dataset.³ In terms of the number of agent turns, there are about 20k / 4k examples in the train / dev set. The blind held-out test set contains 800 examples. WoW Dinan et al. [2019] contains over 20k English social chat conversations with an average of 9 turns on over 1k open-domain topics. For each agent turn, the agent (i.e., wizard) chose one or no grounding sentence from on average 7 Wikipedia passages retrieved by a pre-defined retriever based on the dialogue history for composing the response. Each passage contains 10 sentences. The original data has its dev/test set split to two subsets, which contain conversations about topics seen or unseen in training. It contains about 18k dialogues for training, 2k dialogues for validation and 2k dialogues for test. The test set is split into two subsets, Test Seen and Test Unseen. Test Seen contains 965 dialogues on the topics overlapped with the training set, while Test Unseen contains 968 dialogues on the topics never seen before in training and validation set.

Evaluation Metrics For evaluation, we use exact match (EM) and token-level F1 score as originally used in [Feng et al., 2020; Dinan et al., 2019].

²Doc2Dial: <https://github.com/doc2dial/sharedtask-dialdoc2021>; WoW: https://github.com/facebookresearch/ParlAI/tree/master/parlai/tasks/wizard_of_wikipedia

³Note that the knowledge identification task only targets to predict grounded knowledge for agent turns.

3.3.2 Implementation Details

Doc2Dial Documents are automatically split into passages by parsing the html files into smaller sections with different titles, resulting in an average of 8 passages per document. All passages from a single html file are used throughout each conversation. Knowledge semantic units (SU) are spans as segmented in the data. During inference, we set the maximum knowledge length to be 5 spans based on the dev set distribution.⁴

WoW Passage segmentation is automated in the WoW pre-defined retriever and the dataset provides 7 passages on average for each agent turn in the dataset. Since agents are allowed to select no grounding sentence, we add an additional passage with only one single sentence “no passages used” following the original data processing script.⁵ Knowledge SUs are sentences. We set the maximum knowledge length to be 1 sentence during inference. Passages may differ for each agent turn in the same conversation. Thus, during training, we only calculate history loss for previous agent turns whose ground truth knowledge can be found in next turn passages, which on average cover over 70% of history agent turns.

Experimental Setup We initialize and finetune on BERT [Devlin et al., 2019] downloaded from Huggingface Transformers [Wolf et al., 2020].⁶ We use the uncased base BERT in most of our experiments, and set $3e^{-5}$ as the learning rates and 1000 as warm-up steps. For each experiment, we search the weights in Eq. (3.4) on the dev set in the ranges of $\alpha = \{0.5, 1, 2\}$, $\beta = \{0.5, 2.5, 5\}$. We do not observe much difference with different weight combinations, but the best result is achieved when $\beta = 5$ for both datasets, and $\alpha = 1$ for Doc2Dial and $\alpha = 0.5$ for WoW. We search for fewer than 5 hyperparameter trials for each experiment. All models are trained for 20 and 10 epochs for Doc2Dial and WoW respectively. Models are selected based on the best dev set EM score. The maximum length of dialogue context is 128. The maximum lengths of model input are 512 and 384 for Doc2Dial and WoW respectively, due to the larger variation in passage length in Doc2Dial. In training, we provide multiple passages from the grounding document as the input, where only one of them is the gold passage. We find that learning benefits from having more negative passage examples, and the number used is constrained by memory consumption (up to 10 for the

⁴We use data, baselines, preprocessing & evaluation scripts at: <https://github.com/doc2dial/sharedtask-dialdoc2021>

⁵<https://github.com/facebookresearch/ParlAI>

⁶<https://github.com/huggingface/transformers>

models with posterior regularization and up to 20 otherwise). For inference, up to 20 passages from the target document are considered. For longer documents, the first 20 passages are used. Each training process is run on 2 NVIDIA Quadro Q6000 GPUs. It takes about 18 and 10 hours to train with or without posterior regularization for both datasets. The inference time takes less than 1 minutes per 1000 examples for all experiment settings on 2 GPUs. All our models based on uncased BERT base model contains between 110 to 115 million parameters.

3.3.3 Compared Systems

BERTQA-Token: The original baseline [Feng et al., 2020] and the best published model on Doc2Dial before our work was published. It uses BERTQA [Devlin et al., 2019] with each dialogue context as the question and sliding windows to process each document, and predicts the start and end tokens in the document.

BERTQA-Span: Similar to BERTQA-Token, but predicts the start and end knowledge spans instead of tokens. Instead of using sliding windows, we increase the number of position embeddings to be 2048, initialized with 512 position embeddings in BERT repeated 4 times, following Beltagy et al. [2020]. We observe better results with this operation than when using sliding windows.

Transformer MemNet: The original baseline [Dinan et al., 2019] of WoW, which uses a vanilla Transformer [Vaswani et al., 2017] to encode all knowledge sentences separately and a memory network for sentence selection. Another model version includes pre-training on Reddit conversations.

SLKS: The state-of-the-art model [Kim et al., 2020] on WoW that encodes all knowledge sentences and dialogue turns separately with BERT (or RNN). It uses two GRUs to update the states of dialog history and previously selected sentences.

DiffKS: This model [Zheng et al., 2020] is similar to SLKS. Additionally, it computes the representation difference between each candidate knowledge sentence and the state of previously used knowledge for in the final decision function.

Method	Overall	
	EM	F1
BERTQA-Token	34.6	53.2
BERTQA-Token (our version)	35.8	52.6
DIALKI ($\mathcal{L}_{\text{next}}$ only)	51.2	64.7
DIALKI	59.5	71.0
DIALKI (BERT-large)	61.8	73.1

Table 3.1: Evaluation results on the Doc2Dial test set.

Multi-Sentence: This baseline is designed to be similar to DIALKI, but divides documents into sentences instead of passages. It calculates the next knowledge prediction loss $\mathcal{L}_{\text{next}}$ only. For Doc2Dial, we use subsections, mostly single sentences, as segmented in documents. Knowledge strings rarely exceed the subsection boundaries.

DIALKI (Ours): Our multi-passage knowledge identification model with the next turn knowledge prediction loss $\mathcal{L}_{\text{next}}$, history knowledge prediction loss $\mathcal{L}_{\text{hist}}$, contextualization mechanism (*know-ctx*) and posterior regularization loss \mathcal{L}_{adv} .

3.3.4 Quantitative Results

Doc2Dial Table 3.1 reports the results of different systems in the blind held-out test set with an unseen Covid-19 domain. All models are based on the BERT-base model except the last one that uses BERT-large. The full model of DIALKI achieves best results, demonstrating the effectiveness of combining all components of the system described in § 3.2. The significant improvement from DIALKI ($\mathcal{L}_{\text{next}}$ only) over BERTQA-Token, which takes the full document as a single string, shows the benefit of our multi-passage framework. BERT-large helps further improve the overall results.⁷

WoW Results on both test sets are presented in Table 3.2, containing conversations on seen and unseen topics in training. DIALKI significantly outperforms all other systems, which encode knowledge sentences disjointly. This again confirms the advantage of our multi-passage framework and the modeling of dialogue-document relations.

⁷After ensemble with other large language models of RoBerTa and ELECTRA [Liu et al., 2019; Clark et al., 2020], our model achieves EM / F1 scores as 67.09 / 76.34, achieving the best scoring system on the leaderboard outperforming beating the second-best scores 63.53 / 75.94.

Method	Seen		Unseen	
	EM	F1	EM	F1
Transformer MemNet	22.5	33.2	12.2	19.8
Transformer MemNet + Pretrain	24.5	36.4	23.7	35.8
DiffKS (RNN)	25.5	–	19.7	–
SLKS (RNN)	23.4	–	14.7	–
SLKS (BERT-base)	26.8	–	18.3	–
Multi-Sentence (BERT-base)	30.4	37.7	27.6	35.4
DIALKI (BERT-base)	32.9	40.7	35.5	43.4

Table 3.2: Evaluation results of WoW test sets.

Method	Doc2Dial						WoW					
	Overall		Seen		Unseen		Overall		Seen		Unseen	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
BERTQA-Token	42.2	58.1	48.3	61.1	37.0	55.6	–	–	–	–	–	–
BERTQA-Span	46.3	59.3	54.4	63.5	39.4	55.6	–	–	–	–	–	–
Multi-Sentence	59.5	68.8	63.6	71.6	56.0	66.4	29.2	37.0	32.4	39.7	26.1	34.3
DIALKI ($\mathcal{L}_{\text{next}}$ only)	60.4	71.2	64.2	72.3	57.1	70.2	31.5	39.7	33.3	41.1	29.8	38.3
+ $\mathcal{L}_{\text{hist}}$	63.0	72.6	66.5	73.9	59.9	71.9	33.6	41.6	35.1	42.7	32.2	40.5
+ $\mathcal{L}_{\text{hist}}, \text{know-ctx}$	63.8	73.4	67.7	74.8	60.5	72.3	33.6	41.5	35.2	42.8	32.1	40.3
+ \mathcal{L}_{adv}	64.4	73.8	66.2	73.9	62.8	73.7	32.9	40.8	34.6	42.2	31.1	39.5
+ $\mathcal{L}_{\text{hist}}, \mathcal{L}_{\text{adv}}, \text{know-ctx}$	65.9	74.8	67.6	74.9	64.4	74.7	34.2	42.1	35.9	43.5	32.6	40.7

Table 3.3: Ablation results on Doc2Dial and WoW dev sets.

Surprisingly, DIALKI achieves even higher results on the unseen test set while others observe performance drops. One potential reason is that blindly dividing passages into disjoint sentences to process may hurt the model’s reasoning ability and generalization. In addition, Transformer MemNet, DiffKS and SLKS decouple the encoding of dialogue history and grounding sentences, which prevents the model to effectively reason over their relations. With more investigation into the data, we found that two thirds of the grounding passages used in those conversations about unseen topics actually appear in the training set based on title matching.⁸ Moreover, all topics in the dataset are similar lifestyle topics collected from persona description sentences in [Zhang et al., 2018]. These observations support the possibility of better performance on the original unseen set.

Impact of different components of DIALKI Table 3.3 reports results of ablating different components of our system on the dev set of both datasets. Although Doc2Dial does not provide separate seen and unseen

⁸Only 1.6% of agent turns in the WoW unseen (topic) test set have all of their retrieved passages not seen in training.

KI Model	Knowledge Input	sacrebleu
–	full doc	22.84
BERTQA-Token	pred span	21.42
DIALKI	pred span	25.16
DIALKI	pred span & passage	25.84

Table 3.4: Response generation results on Doc2Dial dev set. KI stands for Knowledge Identification.

Dialogue Context	<p><i>User:</i> I want to trade in my license for a New York one.</p> <p><i>Agent:</i> You have to exchange your-out-state driver license within 30 days of . . .</p> <p><i>User:</i> What if I need my license for when I go back to my other country?</p> <p>. . .</p> <p><i>Agent:</i> Are you studying in New York State?</p> <p><i>User:</i> No</p>	<p><i>User:</i> I heard something about co-op training program. Could you tell me about it, please?</p> <p><i>Agent:</i> A co-op training program refers to . . .</p> <p><i>User:</i> Sounds awesome. What should I do to get that?</p> <p><i>Agent:</i> Well, for that, lets do a little fact check, shall we? Are you using VA educational assistance?</p> <p><i>User:</i> Yes, I’m using that.</p>
Baseline	Do you need a New York State driver license?	Good. In that case, you may be able to get money for books, tuition and housing.
Ours	Are you a driver from another country?	Are you enrolled at an approved institution of Higher Learning?
Gold Response	Are you a licensed driver from another country?	Good. Now, are you enrolled at an approved institution of Higher Learning or IHL?

Table 3.5: Sample generated responses from BART with the full grounding document (baseline) or the predicted grounding span and passage by DIALKI (ours) as the additional input to the dialogue context.

sets as WoW does, we split the dev set into examples that have grounding documents seen or unseen in the training set. Note that “seen” and “unseen” refer to documents and topics for Doc2Dial and WoW respectively.

We observe that DIALKI consistently beats baseline models that either process the full document as a single string or isolated sentences. Our framework leads a to smaller performance gap between seen and unseen examples. Adding the auxiliary history knowledge prediction loss (\mathcal{L}_{hist}) leads to further improvements on both datasets. Adding *know-ctx* helps enhance the performance on Doc2Dial while does not appear to be effective on WoW, as explored further below. Adding posterior regularization (\mathcal{L}_{adv}) is effective on both datasets, while Doc2Dial gets more advantage from it especially on the unseen subset. Combining all model components yields the best results.

Method	Doc2Dial		WoW	
	Seen	Unseen	Seen	Unseen
BERTQA-Span	76.9	72.7	–	–
Multi-Sentence	85.3	81.6	68.0	57.8
DIALKI ($\mathcal{L}_{\text{next}}$ only)	86.6	84.4	72.9	69.0
DIALKI	88.5	87.5	73.4	69.7

Table 3.6: Passage prediction accuracy on dev sets.

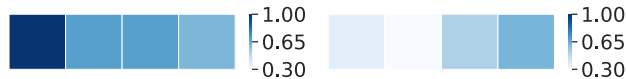


Figure 3.3: Similarity between \mathbf{z} and the latest 4 history turn representations (i.e., $\mathbf{u}_1 \dots \mathbf{u}_4$ from left to right) on Doc2Dial (left) and WoW (right).

3.3.5 Analysis

Impact on Response Generation We apply BART [Lewis et al., 2020] to decode agent responses given the concatenated dialogue context and grounding knowledge (e.g., document or predicted knowledge string) as the input. BART is also used as the baseline for the agent response generation task on Doc2Dial [Feng et al., 2020],⁹ where the model is given the dialogue history concatenated with the full document to decode the next agent response. We conduct experiments on the same model architecture, with the knowledge input being the predicted knowledge string or passage. Without changing the model at all, using knowledge predicted by DIALKI leads to almost 3 points in the sacrebleu score [Post, 2018], as shown in Table 3.4. Examples of generated responses are shown in Table 3.5.

Passage Identification Accuracy We map predicted knowledge strings back to the passages and calculate the passage-level accuracy. Table 3.6 shows that DIALKI outperforms baseline models in locating the passage containing the knowledge string. Notably, our models generalize well in passage prediction to unseen documents or dialogue topics.

Similarity Between Global and History Turn Representations The dot product between \mathbf{z} (the encoding of the whole input sequence) and each history utterance representation \mathbf{u}_i (sigmoid normalized) is used in the gating function g (defined in § 3.2.1) that gates the effect of each utterance u_i in calculating span

⁹<https://github.com/doc2dial/sharedtask-dialdoc2021>

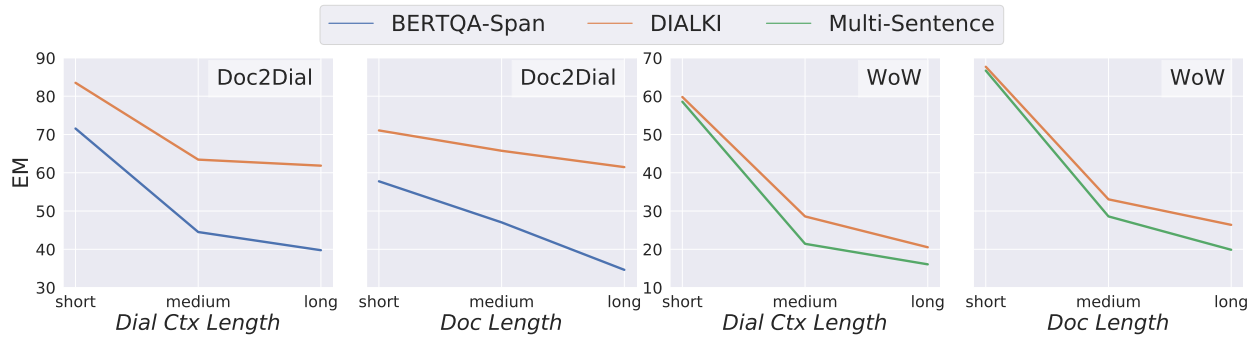


Figure 3.4: EM versus the length of dialogue context (# previous turns) or document (# tokens).

embeddings. Figure 3.3 shows such normalized dot product scores between \mathbf{z} and the latest four history turn vectors. In Doc2Dial, the score is relatively high for the more recent user turn and decreases for earlier turns. Such patterns are not observed in WoW. One potential reason is that each agent in a Doc2Dial conversation has a clear goal to directly address user queries, while WoW conversations are more like social chat. This distinction may explain why *know-ctx* does not work well on WoW. The reason for even higher similarities with earlier turns in WoW could be that knowledge is related to people being referred to with pronouns with names being introduced earlier.

Availability of History Knowledge Labels In Doc2Dial, we calculate the history knowledge prediction loss (\mathcal{L}_{hist}) for all history turns, since all the labels are available in the training set. In practice, it might not be feasible to annotate all history turns, particularly user turns. Hence, we conduct experiments comparing scenarios where history knowledge labels are given for all turns, agent turns only, random 50% agent turns, or no turns. We get EM scores of 63.0, 62.7, 62.4 and 60.4 respectively, finding that removing user turn labels and half of agent turn labels do not affect the results much.

Impact of Dialogue / Document Length Figure 3.4 shows the average EM scores vs. the dialogue context and grounding document length on the Doc2Dial and WoW dev set. Dialogue context lengths are grouped into 0-2 (short), 3-5 (medium) and ≥ 6 (long) history turns. In Doc2Dial, documents are categorized as short, medium and long: 0-500, 501-1000 and 1000+ tokens, respectively. In WoW, documents are categorized as short, medium and long: 0-800, 801-1600 and 1600+ tokens, respectively. DIALKI shows less performance drop as the two input lengths increase compared with baselines that do not leverage the

multi-passage structure of grounding documents.

Span Prediction Error Analysis We randomly select and analyze 50 examples from both Doc2Dial and WoW where DIALKI makes wrong predictions ($EM=0$). Since DIALKI can achieve relatively high passage-level prediction accuracy as shown in Table 3.6, we focus on analyzing prediction errors where the passage is predicted correctly. Among the 50 examples from Doc2Dial, we find 6 error types of predicted knowledge: (1) overlapped with the ground truth knowledge string while including redundant or missing relevant details (38%); (2) relevant to the user query but at an incorrect granularity level (18%); (3) completely irrelevant (16%); (4) relevant to history user queries instead of the current one (14%); (5) contains keywords of the last user utterance that are irrelevant (10%); (6) wrong gold labels (4%).

For WoW, the conversation style is different and the predicted knowledge is a single sentence instead of multiple spans, so prediction errors fall into different classes: (1) open-ended situations where the predicted knowledge is appropriate (48%); (2) unnatural for use in the next response (30%); (3) the predicted knowledge is more appropriate to use than the ground truth (12%); (4) irrelevant knowledge that does not answer the user’s questions (10%). The high percentage of open-ended examples explains the relatively low evaluation scores of knowledge identification on WoW.

3.4 Summary

In summary, we introduce DIALKI to address knowledge identification in conversational systems with long grounding documents, taking advantage of document structure to contextualize document passages together with the dialogue history. DIALKI uses a multi-task objective that identifies knowledge for the next turn and used knowledge for previous turns, which captures interconnections between the dialogue and the document. Additional posterior regularization in learning further improves results. The model gives state of the art performance for this task on Doc2Dial and Wizard of Wikipedia, respectively. We show that improvements in knowledge selection transfer to response generation with a baseline generator.

The current study is limited by the static nature of the available data. Further work is needed to assess performance in an interactive setting. In Chapter 5, we propose a task and models for information-seeking conversations that consider a more interactive and realistic scenario; e.g., the user intent is ambiguous or no

direct answer can be found in the document.

Ethical Considerations

Our work is primarily intended to address and encourage future work on the task of knowledge identification in document-grounded dialogue systems. We believe that locating relevant knowledge to be used in the following conversation can be useful for improving the interpretability of response generation models. Knowledge identification can also play an important role in human-in-the-loop assistant scenarios. This places greater control into the hands of human agents instead of automatic response generation models, which tend to suffer from ethical issues like generating hallucinated [Zellers et al., 2019; Wu et al., 2021c] or toxic content [Pavlopoulos et al., 2020].

Chapter 4

CONQRR: Conversational Query

Rewriting for Retrieval with Reinforcement Learning

4.1 Introduction

In DIALKI (Chapter 3), the goal is to identify specific knowledge in a given document that is relevant given the current dialogue context. In contrast, this chapter addresses the problem of identifying a relevant passage in a document collection. Passage retrieval in an open-domain conversational question answering (CQA) system [Anantha et al., 2021], compared to standard retrieval tasks [Voorhees and Tice, 2000; Bajaj et al., 2018], poses new challenges of understanding user questions within the dialogue

context. Most existing conversational retrieval models [Yu et al., 2021; Lin et al., 2021; Kim and Kim, 2022] rely on training specific retrievers like dual encoders [Karpukhin et al., 2020]. However, re-training

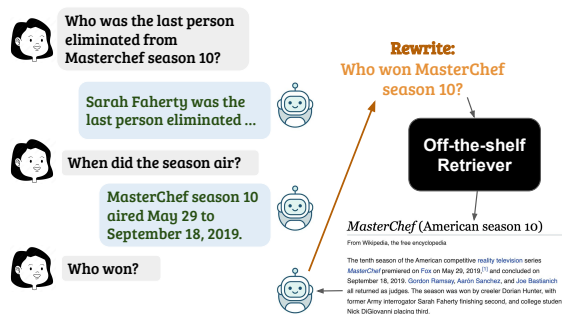


Figure 4.1: A CQA agent rewrites the current user question into a more effective one (in orange) for the given *off-the-shelf* retriever to find the most relevant passage.

well-established retrievers for conversational queries can be expensive or even infeasible due to their complicated system designs (e.g., those used in search engines). Moreover, the preference and availability of such off-the-shelf retrievers can vary depending on the end users.

The task of question-in-context rewriting or query rewriting (QR) in a conversation [Elgohary et al., 2019; Dalton et al., 2020] is to convert a context-dependent question into a self-contained question. It enables the use of any off-the-shelf retriever, which we define as a retriever that cannot be fine-tuned or provide access to any of its internal architecture design or intermediate results (i.e., can only be seen as a black-box).

Therefore, in this chapter, we focus on *query rewriting* for the task of *conversational passage retrieval* in a CQA dialogue with *any off-the-shelf* retrieval system that can only be used as a black box. Specifically, we seek to build a QR model that rewrites a user query into the input of the retriever, in such a way that optimizes for passage retrieval performance. Figure 4.1 shows an example of our task, where given an off-the-shelf retriever, the agent rewrites the current user query “Who won?” into a more effective query for retrieval.

Recent work that leverages QR for conversational passage retrieval [Anantha et al., 2021; Dalton et al., 2020] collects human-rewritten queries to train a supervised QR model. However, humans are usually instructed to rewrite conversational queries to be unambiguous to a human outside the dialogue context, which does not necessarily align with *the goal in our task*—to optimize the retrieval performance. Our experiments will show that these human rewrites indeed sometimes omit information from the dialogue context that is useful to the retriever. This limitation of human query rewrites impacts supervised training. In addition, prior supervised QR models are agnostic to downstream retrievers as they are separately trained using human rewrites that summarize the context from a human understanding perspective, i.e. without considering what might be most effective for a retriever.

We propose a reinforcement learning (RL)-based model CONQRR (**C**onversational **Q**uery **R**ewriting for **R**etrieval). It directly optimizes the rewritten query towards retrieval performance, using only weak supervision from retrieval. We adopt a novel reward function that computes an approximate but effective retrieval performance metric on in-batch passages at each training step. Our reward function does not assume any specific retriever model design, and is generic enough for CONQRR to adapt to any off-the-shelf

retriever.

We show CONQRR outperforms existing QR models on a recent large-scale open-domain CQA dataset QReCC [Anantha et al., 2021] by over 12% and 14% for BM25 and a neural dual encoder retriever model [Ni et al., 2022] respectively, averaging over three retrieval metrics. We observe the performance boost on all three QReCC subsets from different conversation sources, including one that only appears in the test set (i.e., out-of-domain).

This chapter contains material that was originally published in Wu et al. [2022]. I was the main contributor of this work, and discussed and wrote the paper with the other collaborators throughout the project.

4.2 Approach

Problem Definition We focus on the task of *query rewriting (QR)* for *conversational passage retrieval* in a CQA dialogue, with an *off-the-shelf* retriever. The task inputs include a dialogue context x consisting of a sequence of previous utterances $(u_1, u_2, \dots, u_{n-1})$, the current user question u_n , a passage corpus P and an off-the-shelf retriever R .¹ R cannot be fine-tuned but returns a ranked list of top-k passages when given a query string and a passage corpus, and no other assumption about the model architecture of R can be made. The task aims to rewrite x into a query q such that R can take q as the input query to retrieve passages relevant to x from P . Specifically, a passage p is relevant to x if p provides enough information to answer u_n in the context of $(u_1, u_2, \dots, u_{n-1})$.

In this section, we first describe a supervised QR model based on T5 (T5QR) [Lin et al., 2020a] that applies a generic Seq2Seq training objective with QR labels (§ 4.2.1). Then we introduce our RL-based framework CONQRR (**C**onversational **Q**uery **R**ewriting for **R**etrieval) that trains a QR model to optimize towards retrieval and is adaptable to any given off-the-shelf retriever, with weak retrieval supervision (§ 4.2.2).

4.2.1 T5QR

T5 is an encoder-decoder model that is pre-trained on large textual corpora [Raffel et al., 2020]. Following Lin et al. [2020a], we fine-tune T5 to rewrite a conversational query with the input as the concatenation of utterances in the dialogue context x and the output as the human rewrite \hat{q} . Note that we concatenate

¹To mimic practical use cases, R is usually assumed to be general purpose retriever with standard search queries.

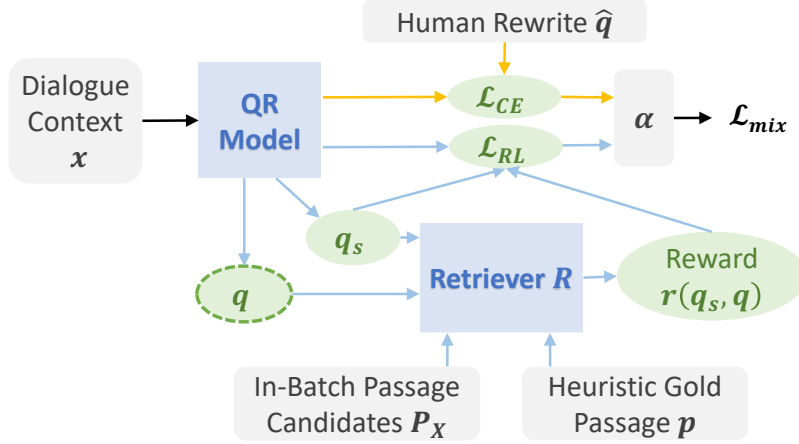


Figure 4.2: Our CONQRR framework. Yellow and blue arrows mark the flow of CE (**unused when** $\alpha = 1.0$) and RL loss calculation, respectively. During inference, only q (dashed border) is generated as the final rewrite.

the utterances in a reversed order such that u_n becomes the first one in the input string and any truncation impacts more distant context. Utterances are separated with a separator token “[SEP]” in the concatenated string. The model is then trained with a standard cross entropy (CE) loss to maximize the likelihood of generating \hat{q} , which is a self-contained version of the query u_n that can be interpreted without knowing previous turns $(u_1, u_2, \dots, u_{n-1})$ in x .

4.2.2 CONQRR

QR models trained with a standard CE loss are agnostic to the retriever. In addition, human rewrites are not necessarily the most effective ones for passage retrieval (see § 4.3.3 for an exploration).

This motivates us to design our RL-based framework CONQRR (Figure 4.2) that trains a QR model directly optimized for the retrieval performance and can be adapted to any given off-the-shelf retriever. Here, the RL environment includes the retriever model, dialogue context and passage candidates, in which the QR model takes actions by generating rewritten queries and obtains rewards accordingly.

To be comparable with supervised QR models that do not use gold passages in training, we first describe how we obtain weak retrieval supervision for the RL reward calculation in CONQRR. Then we introduce the RL training details of CONQRR.

Weak Retrieval Supervision In a CQA dialogue, each question naturally comes with an answer in its following conversational utterance. For each x , we mark its weak passage label p as the one having a string span with the highest token overlap score with the following answer string u_{n+1} :

$$p = \arg \max_{p' \in P} \left[\arg \max_{s \in p'} \text{sim}(s, u_{n+1}) \right] \quad (4.1)$$

where s is a string span in the passage p' and $\text{sim}()$ calculates the token overlap score as F1 assuming u_{n+1} is the target and s is the prediction.² Tokens are lower-cased from the NLTK tokenizer.³ However, as searching within all candidates in P is very time-consuming, we instead first use BM25 to retrieve the top 100 passages from P with the BM25 input being the human rewrite,⁴ and then locate the best passage p from these 100 candidates.

RL Training CONQRR also has T5 as the base model architecture, but it can also be initialized with T5QR, which we explore in our analysis in § 4.3.

For each training example with the dialogue context x , we use the concatenated utterances in x as the model input. For each input, we generate m sampled rewritten queries $(q_{s_1}, \dots, q_{s_m})$ as well as a baseline generated rewrite q . To generate each sampled rewrite q_s , at time step t of the decoding process, a token q_s^t is drawn from the decoder probability distribution $Pr(w|x, q_s^{1:t-1})$. The baseline rewrite q is the output of greedy decoding,⁵ which is also applied for query rewriting during inference. We then apply a self-critical sequence training algorithm [Rennie et al., 2017] to calculate the reward for each q_s relative to q as $r(q_s, q) = \text{score}(q_s) - \text{score}(q)$. The intuition is to reward/penalize the generation of sampled rewrites that lead to better/worse retrieval performance than greedy decoding used during inference. Ideally, the $\text{score}()$ function should be some retrieval evaluation metric like mean reciprocal rank (MRR) or Recall@K. However, as it is very costly to run actual retrieval for each training step, we instead use an approximate scoring function described below.

To compute $\text{score}(\bar{q})$ for a rewrite \bar{q} , we first use \bar{q} to do retrieval from the in-batch passage candidates

²We randomly choose a passage if there is a tie in scores.

³<https://www.nltk.org>

⁴We show in § 4.3.3 that using the dialogue context as the BM25 input to induce weak supervision gives similar performance (Figure 4.3), where no human rewrites are used.

⁵We tried beam search with various beam sizes and got similar results as greedy decoding.

P_X defined as follows, instead of from the full passage corpus P . We pre-compute one positive and one negative passage (p and p_n) for each training example x where p_n is a randomly selected passage that is different from p , 50% of the time from the top 100 BM25-retrieved candidates as hard negatives (with the BM25 input being the human rewrite) and remaining 50% of the time from P . We define the set of all such positive and negative passages of input examples in a batch X as the in-batch passage candidates P_X . Formally, we define $P_X = \{p^i, p_n^i | x_i \in X\}$ as the set of in-batch passage candidates for the batch X . Then for a generated rewritten query \bar{q} of $x \in X$, we calculate $score(\bar{q})$ as a binary indicator of whether the retriever R ranks the assigned positive passage p highest from P_X . We denote $R(\bar{q}, P_X, k)$ as the k -th most relevant passage retrieved by R from the candidate pool P_X , and define:

$$score(\bar{q}) = \mathbb{1}[R(\bar{q}, P_X, 1) = p] \quad (4.2)$$

Then the RL training loss for x becomes:

$$\begin{aligned} \mathcal{L}_{RL} &= -\frac{1}{m} \sum_{i=1}^m r(q_{s_i}, q) \log Pr(q_{s_i} | x) \\ Pr(q_{s_i} | x) &= \prod_{t=1}^{|q_{s_i}|} Pr(q_{s_i}^t | x, q_{s_i}^{1:t-1}) \end{aligned}$$

Following prior work [Paulus et al., 2018; Celikyilmaz et al., 2018], we experiment with a pure RL loss (\mathcal{L}_{RL}) and a mixed RL and CE loss in training:

$$\mathcal{L}_{mix} = \alpha \mathcal{L}_{RL} + (1 - \alpha) \mathcal{L}_{CE} \quad (4.3)$$

where $\alpha \in [0, 1]$ is a tunable parameter.

Inference At inference time, both T5QR and CONQRR work in the same way. The trained QR model greedily generates the rewritten query given a dialogue context. Then, the predicted rewrite is given to the provided retriever to perform retrieval.

4.2.3 Retriever Models

We evaluate the effectiveness of CONQRR in experiments with two general-domain retrieval systems.

BM25 We follow Anantha et al. [2021] using Pyserini [Yang et al., 2017] with default parameters $k1 = 0.82$ and $b = 0.68$. These values were chosen based on retrieval performance on MS MARCO [Bajaj et al., 2018], which contains non-conversational queries only. During the RL training of DIALKI, due to the complexity of applying Pyserini to calculate rewards on-the-fly, we instead use a Pyserini approximate called BM25-light. The only differences between them are that BM25-light (1) uses T5’s subword tokenization instead of whole word tokenization and (2) does not use special operations (e.g., stemming) as applied in Pyserini. After training, we still run inference and report retrieval performance on BM25. Pyserini simply encodes the whole query input and each passage without truncating. We set maximum query and passage length as 128 and 2000 for BM25-light, but only less than 0.1% cases require truncation with these thresholds.

Dual Encoder (DE) We use a recent T5-base dual encoder model [Ni et al., 2022] which achieves state-of-the-art performance on multiple retrieval benchmarks. This model is fine-tuned on MS MARCO, and kept fixed for our experiments. The maximum query or passage length is 384. The average passage length is 378, but we observe performance drop by further increasing the maximum length for the dual encoder.

4.3 Experiment

Dataset QReCC [Anantha et al., 2021] is a dataset of 14k open-domain English conversations in the format of alternating user questions and agent-provided answers with 80k question and answer pairs in total. The conversations are collected from different sources: QuAC [Choi et al., 2018], Natural Questions [Kwiatkowski et al., 2019] and TREC CAsT-19 [Dalton et al., 2020] with additional annotations by crowd workers. QReCC reuses questions in QuAC and TREC conversations and re-annotates answers. For each NQ-based conversation, they only use one randomly chosen question from NQ to be the starting question and then annotate the remaining conversation. Therefore, QReCC can be divided into three subsets for evaluation. We name them as *QuAC-Conv*, *NQ-Conv* and *TREC-Conv* respectively to differentiate them

from the original datasets from which they are derived. In total, there are 63k, 16k and 748 question and answer pairs in QuAC-Conv, NQ-Conv and TREC-Conv respectively, where TREC-Conv only appears in the test set. The original data is only divided into train and test sets. We randomly choose 5% examples from the train set to be our validation set. Each user question comes with a human-rewritten query. For each agent turn, gold passage labels are provided if any. In some conversations from QuAC-Conv, the first user query is ambiguous as it depends on some topical information from the original QuAC dataset. Therefore, in order to fix this issue, we follow Anantha et al. [2021] to replace all first user queries in QReCC conversations with their corresponding human rewrites.

QReCC is a publicly available dataset that was released under the Apache License 2.0 and we use the same task set-up proposed by the original QReCC authors. The entire text corpus for retrieval contains 54M passages, segmented in the released data.⁶

Evaluation Metrics Following Anantha et al. [2021], we use mean reciprocal rank (MRR), Recall@10 and Recall@100 to evaluate the retrieval performance by using the provided evaluation scripts.⁷ We use their *updated* evaluation script for most experiments, except that we also use the *original* version for calculating scores in Table 4.2 to compare with their reported QReCC baseline results. We note that these two evaluation scripts only differ by a scaling factor⁸ so they should lead to the same conclusions regarding model comparisons.

Implementation Details Our models are implemented using JAX.⁹ For training, we set 64, 1k and 10k as the batch size, warm-up steps and total training steps, respectively. We use e^{-3} and e^{-4} as the learning rate for T5QR and DIALKI, respectively. We use Adafactor [Shazeer and Stern, 2018] as our optimizer with the default parameters. Linear decay is applied after 10% of the total number of training steps, reducing the learning rate to 0 by the end of training. Following prior work on RL for text generation [Paulus et al., 2018; Fisch et al., 2020], we first initialize CONQRR with a supervised model (T5QR) [Lin et al., 2020a] as a warm-up. Our RL optimization (self-critical sequence training [Rennie et al., 2017]) uses a policy

⁶Original QReCC data: <https://zenodo.org/record/5115890#.YZ8kab3MI-Q>.

⁷Both original and updated evaluation scripts: <https://github.com/scal-conf/SCAI-QReCC-21>.

⁸This is due to the exclusion of test examples with no valid gold passage labels (roughly 50%) in the updated evaluation, which results in 6396, 1442 and 371 test instances for QuAC-Conv, NQ-Conv and TREC-Conv, respectively.

⁹<https://github.com/google/jax>

QR Model	IR System	QReCC (Overall)			QuAC-Conv			NQ-Conv			TREC-Conv (OOD)*		
		MRR	R10	R100	MRR	R10	R100	MRR	R10	R100	MRR	R10	R100
T5QR	BM25	0.328	52.5	84.7	0.33	52.7	85.0	0.345	54.2	83.9	0.230	44.5	82.3
DIALKI (mix)	BM25	0.373	58.5	90.2	0.379	59.2	90.9	0.385	58.8	88.9	0.229	44.7	82.7
DIALKI (RL)	BM25	0.383	60.1	88.9	0.395	61.6	90.2	0.378	58.0	86.7	0.198	43.5	75.9
Human Rewrite	BM25	0.398	62.6	98.5	0.403	62.9	98.4	0.408	63.8	99.0	0.273	53.8	98.9
T5QR	DE	0.361	56.2	75.9	0.349	55.7	76.1	0.417	58.7	74.2	0.343	55.9	79.2
DIALKI (mix)	DE	0.395	61.9	81.8	0.387	62.0	82.4	0.439	62.2	79.0	0.361	58.9	81.0
DIALKI (RL)	DE	0.418	65.1	84.7	0.416	65.9	85.8	0.453	64.1	80.9	0.327	55.2	79.6
Human Rewrite	DE	0.422	64.8	84.0	0.409	64.5	84.1	0.483	65.8	83.2	0.411	66.0	86.5

Table 4.1: Passage retrieval performance on QReCC test set and 3 subsets. DIALKI (mix) beats the supervised T5QR model on all retriever system and test set combinations. * OOD (out-of-domain): only appear in the test set.

gradient method with Monte Carlo sampling. In § 4.3.2, our models use T5-base as the base model, and in § 4.3.3 we explore initializing with T5QR showing the potential for some additional performance gains. We experiment with CONQRR trained with either a mixed (\mathcal{L}_{mix}) or pure RL (\mathcal{L}_{RL}) loss. For the mixed loss, we observe that CONQRR works well when the RL loss weight α is large.¹⁰ We tune its values in 0.9, 0.95, 0.97, 0.99, and use 0.99 as the final value. For supervised training, models are selected based on the best dev set Rouge-1 F1 score with the human rewrites, following Anantha et al. [2021]. For RL-based training of DIALKI, models are selected based on the average in-batch gold passage prediction accuracy as in Eq. (4.2) on dev set with greedily decoded rewrites. For the experiment with the pure RL loss and the retriever BM25, our results are obtained with the initialized T5QR model being fine-tuned with only 10% QR labels, as we find initializing with a model using 100% QR labels is unstable for BM25. Previous work [Wu et al., 2021b] also had a similar observation that initializing with a less trained model leads to more stable RL training.

The maximum length of the dialogue context fed into the QR model is 384 (longer than 97.9% dialogue contexts in QReCC) and the maximum output rewrite length is 64 (longer than 99.9% human rewrites). To generate each sampled rewrite q_s (see § 4.2.2), we apply top-k sampling where $k = 20$. For each training example, we sample 5 rewrites in total (i.e., $m = 5$ for the RL training explained in Section 4.2.2). Each training process is run on 8 TPU nodes. It takes about 2 and 9 hours for the supervised and RL-based training, respectively. For each experiment, we observe similar performance or training curves for 2-3 runs and report numbers on a random run. Both T5QR and DIALKI are based on T5-base and have about 220M

¹⁰We also experiment with $\alpha = 0.0$, where the RL loss is removed for both retrievers, and get similar results as T5QR.

parameters.

4.3.1 Compared Systems

For QR models, we compare three supervised models including **GPT2 with weak supervision (WS)** [Yu et al., 2020], a GPT2-medium based system that additionally leverages search sessions to create weak supervision for QR training before fine-tuning, **T5QR** [Lin et al., 2020a] and **Transformer++**, the previous state-of-the-art model based on GPT2-medium [Vakulenko et al., 2021] and reported in the original dataset paper [Anantha et al., 2021], as well as **CONQRR (mix/RL)** with a mixed (\mathcal{L}_{mix}) or pure RL (\mathcal{L}_{RL}) loss. For analysis purposes, we also report performance for directly using the concatenated dialogue context as the retriever input without any query rewriting in § 4.3.3. We experiment with two off-the-shelf retrievers, **BM25** and **DE** (§ 4.2.3).

4.3.2 Quantitative Results

To have a direct comparison with the original QR baseline Transformer++, which has the retrieval performance reported on the overall QReCC test set by using BM25 as the off-the-shelf retriever, we first compare all QR models in the same setting in Table 4.2 and use both the original and updated versions of the provided evaluation script. GPT2 + WS has similar performance as Transformer++. T5QR and CONQRR outperform the Transformer++ baseline by 5% and 18% respectively, averaged on three metrics,¹¹ although Transformer++ is based on a larger base model - GPT2-medium. Therefore, CONQRR (RL) becomes the *state-of-the-art* QR model for conversational passage retrieval on QReCC with the original BM25 retriever in Anantha et al. [2021].

Table 4.1 shows more comprehensive retrieval results comparing CONQRR and the supervised model T5QR, with the updated evaluation script. For the overall QReCC test set, CONQRR outperforms T5QR for all three metrics. For MRR and Recall@10, gains are roughly 15% with the RL loss and 9-14% with the mixed loss for both retrievers. Gains in Recall@100 vary more (4-12%). Breaking down the results by subset shows that the mixed loss is more robust. CONQRR (RL) is less effective for the TREC-Conv subset, which only appears in the test set. This suggests that RL loss alone does not generalize well to out-

¹¹We obtained prediction results from the authors and reran their evaluation script. The numbers we got are slightly lower than what they reported, but do not affect the conclusions.

QR Model	Original Eval			Updated Eval		
	MRR	R10	R100	MRR	R10	R100
GPT2 + WS	0.152	24.7	41.5	0.304	49.6	83.1
Transformer++	0.155	24.8	40.6	0.311	49.8	81.4
T5QR	0.164	26.2	42.3	0.328	52.5	84.7
DIALKI (mix)	0.186	29.2	45.0	0.373	58.5	90.2
DIALKI (RL)	0.191	30.0	44.4	0.383	60.1	88.9
Human	0.199	32.8	49.4	0.398	62.6	98.5

Table 4.2: Passage retrieval performance of QR models, comparable to scores in Anantha et al. [2021] by using the same BM25 retriever for QReCC test set. DIALKI achieves *state-of-the-art* results. Recall@10 and Recall@100 are abbreviated as R10 and R100.

of-domain examples. Across all subsets, the best MRR and Recall@10 results are consistently from DE, whereas BM25 has better Recall@100 scores. Although previous work [Karpukhin et al., 2020] shows that DE retrievers generally lead to better recall scores than BM25, one potential reason for a lower Recall@100 score of DE is that we use an *off-the-shelf* retriever for our retrieval task while most previous work that compares BM25 and DE focuses on fine-tuning the DE model. Without being fine-tuned, a DE model may be more vulnerable to domain shift than BM25. On the other hand, prior work [Luan et al., 2021] shows that a DE model’s performance would drop as the passage length increases. In the QReCC dataset, the average passage length is 378, which is relatively long according to [Luan et al., 2021].

4.3.3 Analysis

Zero or Few QR Supervision We investigate how sensitive CONQRR and T5QR are to the availability of QR labels. We experiment with training T5QR with 0%, 1%, 10% or 100% of QR labels in the QReCC train set. For the case of 0% examples, we simply use the original T5 checkpoint without fine-tuning. When training CONQRR, we mask out the CE loss in Eq. (4.3) for unused QR labels in training its initialized T5QR model, and we use the concatenated dialogue context as the BM25 input to obtain weak gold and hard negative passages for each training example, instead of using human rewrites. Figure 4.3 plots the curve of MRR on the overall QReCC test data using DE as the retriever versus the percentage of QR labels used for training. We see that CONQRR can already significantly outperform T5QR with even 0% or 1% of QR supervision.

The 0% QR label result for RL is similar to RL result in Table 4.1 (≈ 0.42 MRR), showing that using

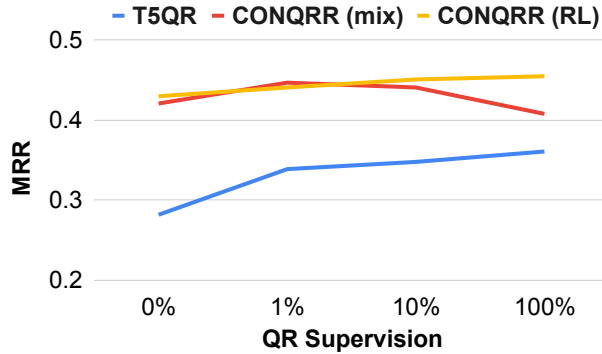


Figure 4.3: MRR on QReCC versus the percentage of QR supervision used for training, with DE as the retriever.

the dialogue context is as effective as the QR label for choosing positive and hard negative passages for RL training. Performance of the RL and mixed loss are similar when there is little supervision, roughly tracking the trends of the T5QR model that it is initialized with. The small difference between training with the mixed and RL loss with 0% QA label comes from their difference in α in Eq. 4.3. The finding that performance degrades for the mixed loss with 100% supervision may be due to a mismatch in the CE and RL losses as minimizing the CE loss does not directly optimize the retrieval performance. Training with RL reaching the best performance with 100% QR labels shows that initializing with T5QR gives a benefit. T5QR is more sensitive to QR supervision but also does not require many QR labels for training, as its curve becomes flattened after 1% supervision.

Effects of Topic Shift & Human Rewrites We hypothesize that a context involving a topic shift will present the greatest challenges for conversational passage retrieval. To explore this factor, we split the QReCC data into topic-concentrated and topic-shifted subsets as follows. A test example (with at least one previous turn) is considered *topic-concentrated* if the gold passage of the current question comes from a document that was used in *at least one* previous turn. In contrast, a test example (with at least one previous turn) is considered *topic-shifted* if the gold passage of the current question comes from a document that was *never* used in any previous turn. There are about 4.7k and 1.1k examples in the topic-concentrated and topic-shifted subsets, respectively. We compare the retrieval performance of different retriever inputs: dialogue context (which uses the concatenated dialogue history without QR), the predicted rewrite from T5QR and CONQRR with two loss alternatives, and the human rewrite. Table 4.3 shows that the dialogue

Input	IR	Topic-Concentrated			Topic-Shifted		
		MRR	R10	R100	MRR	R10	R100
Dial Context	BM25	0.620	81.4	94.9	0.154	39.1	68.6
T5QR	BM25	0.352	54.4	84.0	0.252	45.1	79.1
DIALKI (mix)	BM25	0.419	63.1	91.2	0.252	45.9	82.1
DIALKI (RL)	BM25	0.444	66.2	90.3	0.233	44.5	78.4
Human Rewrite	BM25	0.440	66.7	98.8	0.318	56.7	98.4
Dial Context	DE	0.551	78.1	93.2	0.179	35.7	61.4
T5QR	DE	0.353	55.7	75.4	0.329	50.8	69.2
DIALKI (mix)	DE	0.404	63.8	83.4	0.334	53.2	72.6
DIALKI (RL)	DE	0.445	69.3	87.8	0.303	50.4	73.3
Human Rewrite	DE	0.424	65.5	84.5	0.397	61.0	79.8

Table 4.3: Performance of using different retriever inputs for *Topic-Concentrated* or *Topic-Shifted* examples.

context outperforms even the human rewrite on the topic-concentrated set by 22% and 17%, averaging over three metrics, for BM25 and DE respectively, which shows the *limitation of human rewrites*. We also see that CONQRR (RL) surpass the human rewrite on the topic-concentrated set on MRR for BM25 and all three metrics for DE.

However, for the topic-shifted set, the human rewrite outperforms the dialogue context by 52% and 61%, averaging over three metrics, on BM25 and DE, respectively. The predicted rewrite by CONQRR (mix) outperforms the dialogue context by 30% and 44% on BM25 and DE, respectively. Therefore, compared with dialogue context, QR has great value in the aspect of *robustness to topic shifts*. When comparing with human rewrites, we also see room for improvement for QR models.

These observations are *largely unexplored* in previous work, and they motivate our work on the task of QR for conversational passage retrieval in general, and optimizing directly towards retrieval.

Effect of Dialogue Context Length Figure 4.4 shows the MRR score on topic-concentrated and topic-shifted subsets with DE as the retriever for various dialogue context lengths. Dialogue context lengths are grouped into 1-2, 3-4 and ≥ 4 previous utterances (including the current question). For topic-concentrated conversations, all compared models have similar robustness to the dialogue context length and CONQRR (mix) is slightly more robust than T5QR. For topic-shifted conversations, both QR models and human rewrites show little drop or even an increase in performance as the context length gets longer. In contrast, the robustness of the dialogue context worsens with longer contexts, which confirms the importance

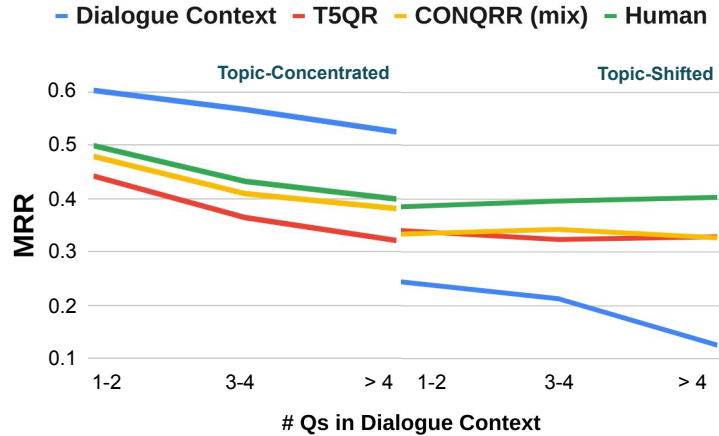


Figure 4.4: MRR versus the number of questions in the dialogue context, with DE as the retriever.

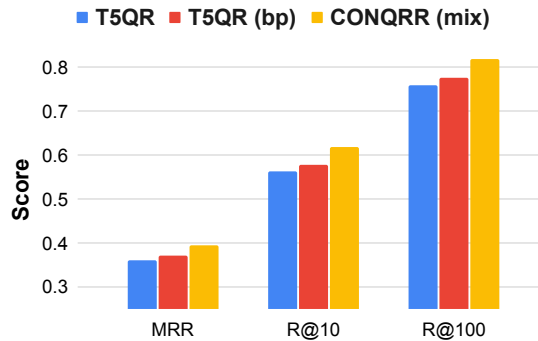


Figure 4.5: Evaluation scores on QReCC for T5QR w/ or w/o brevity penalty and DIALKI (mix), with DE as the retriever. Recall scores (R@k) are divided by 100.

of QR discussed above. We have similar observations for other metrics as well as for the BM25 retriever.

Quantitative Attributes of Rewrites Table 4.4 shows the average number of tokens per rewrite, and the percentage of overlapping tokens (excluding stopwords) between the rewrite and the gold passage(s). CONQRR generally generates longer rewrites with more overlapping tokens with gold passage(s), compared with T5QR. With DE as the retriever, CONQRR (RL) generates more than double the length of T5QR, CONQRR (mix) and even human rewrites.

We hypothesize that simply generating a longer rewritten query is not the only factor that contributes to better retrieval performance. We investigate this by applying a brevity penalty [Wu et al., 2016] during decoding for T5QR such that its average query length matches that of DIALKI (mix). Figure 4.5 shows that DIALKI (mix) still outperforms T5QR with the brevity penalty for all three evaluation metrics on QReCC.

QR Model	QuAC-Conv		NQ-Conv		TREC-Conv	
	L	% OL	L	% OL	L	% OL
T5QR	10.9	35.8	8.9	40.4	8.2	37.8
Ours (mix) w/ BM25	12.1	37.2	9.5	42.1	8.5	38.8
Ours (RL) w/ BM25	11.2	40.2	10.1	44.6	9.4	39.4
Ours (mix) w/ DE	12.1	37.2	9.6	41.7	8.7	39.1
Ours (RL) w/ DE	28.2	51.1	21.7	55.8	18.3	44.3
Human	12.1	37.2	9.3	43.0	8.4	41.7

Table 4.4: Average number of tokens (L) and the percentage of overlapping tokens (OL) with the gold passage(s) in output rewrites.

Dialogue Context	<p><i>Q:</i> What were John Stossel's most popular publications?</p> <p><i>A:</i> Give Me a Break: How I Exposed Hucksters, Cheats, and Scam Artists and Became ...</p> <p>...</p> <p><i>Q:</i> What was the response?</p>	<p><i>Q:</i> What were some notable live performances at the Buena Vista Social Club?</p> <p><i>A:</i> Ibrahim Ferrer and Rubén González ...</p> <p>...</p> <p><i>Q:</i> What other live performances are important?</p>
Gold Passage	<p>Stossel has written three books. Give Me a Break: ...<i>It was a New York Times bestseller for 11 weeks</i> ...</p>	<p>The first performances ... Ibrahim Ferrer and Rubén González performed together ...<i>a 1999 Miami performance</i> ...</p>
DIALK1 (mix)	<p>What was the response to John Stossel's book, Give Me a Break? (Rank=2)</p>	<p>What other live performances at the Buena Vista Social Club are important besides Ibrahim Ferrer and Rubén González? (Rank=2)</p>
T5QR	<p>What was the response to the book Give Me a Break? (Rank >100)</p>	<p>What other live performances are important at the Buena Vista Social Club? (Rank=18)</p>
Human	<p>What was the response to Give Me a Break: How I Exposed Hucksters, Cheats, and Scam Artists and Became the Scourge of the Liberal Media? (Rank >100)</p>	<p>What other live performances of the Buena Vista Social Club are important? (Rank=17)</p>

Table 4.5: Examples of predicted rewrites and the gold passage ranks by using them as the DE retriever input. *The gold answer is italicized in the gold passage.*

Rewrite Quality Analysis and Examples In order to understand why rewrites generated by CONQRR lead to better retrieval performance and even sometimes outperform human rewrites,¹² we sampled 50 examples where CONQRR (mix) leads to better ranking of gold passages than human rewrites (using DE retriever). We notice that 70% of CONQRR generated rewrites contain additional context and (correct) information when compared to human rewrites. The remaining 30% contain alternative or less context information than human rewrites. In such cases, potentially because the information in human rewrites is less relevant to gold passages, it led to a lower gold passage rank. Overall, these CONQRR rewrites are as fluent as human rewrites and contain no major misinterpretation of the dialogue context. Table 4.5 shows

¹²This is only for analysis purposes. Note that the goal of our predicted rewrites is to improve retrieval performance instead of directly being used by end users.

Dialogue Context	<p><i>Q</i>: What is Get 'Em Girls? A: Jessica Mauboy's second studio album, Get 'Em Girls (2010). ... <i>Q</i>: Did she receive any awards or honors during these years?</p>	<p><i>Q</i>: What is one actress who was a Bond girl? A: Ursula Andress in Dr. No is widely regarded as the first Bond girl. <i>Q</i>: Who was another Bond girl?</p>
Gold Passage	<p>... Mauboy performed "Get 'Em Girls" at the 2010 ... received <i>her first nomination for Young Australian of the Year</i> ...</p>	<p>... Ursula Andress (as Honey Ryder) in Dr. No (1962) is widely regarded as the first Bond girl, although she was preceded by both <i>Eunice Gayson</i> as Sylvia Trench and ...</p>
DIALKI (mix)	<p>Did Jessica Mauboy receive any awards or honors during the years she released Get 'Em Girls? (Rank=7)</p>	<p>Who was another Bond girl besides Ursula Andress in Dr. No? (Rank=7)</p>
T5QR	<p>Did Jessica Mauboy receive any awards or honors during these years? (Rank >100)</p>	<p>Who was another Bond girl? (Rank=68)</p>
Human	<p>Did Jessica Mauboy receive any awards or honors during the 2010s? (Rank=24)</p>	<p>Who was another Bond girl, besides Ursula Andress? (Rank=12)</p>

Table 4.6: Examples of predicted rewrites and the gold passage ranks by using them as the **BM25 retriever** input. *The gold answer is italicized in the gold passage.*

Dialogue Context	<p><i>Q</i>: What did Jan Howard do in the early 60s? A: In 1960, Jan Howard went to Nashville, Tennessee, where they appeared on The Prince Albert Show, the Grand Ole Opry segment carried nationally by NBC Radio. <i>Q</i>: Did she get a record deal?</p>
DIALKI (mix)	<p>Did Jan Howard get a record deal? (Rank=69)</p>
Human	<p>Did Jan Howard get a record deal in 1960 after her appearance on The Prince Albert Show? (Rank=6)</p>

Table 4.7: Error analysis example 1: DIALKI (mix) rewrite contains less context than the human rewrite, which leads to worse ranking of the gold passage.

two examples of generated rewritten queries of T5QR and CONQRR (mix) trained with DE in the loop, as well as the human rewrites. In the left example, the CONQRR rewrite includes an entity “John Stossel” that is mentioned in the gold passage but not included by rewrites from T5QR or Human. Thus, even if the human rewrite is longer by containing the book’s full name, CONQRR enables more efficient retrieval with a partial book name along with its author name. In the right example, CONQRR generates a longer rewrite containing richer contextual information. We have similar observations for BM25 and put predicted rewrites from DIALKI (mix) that is trained towards BM25 in Table 4.6. Gold passage ranks are shown in the table, using the predicted rewrites as the BM25 retriever input.

For error analysis, we sampled another 50 examples where CONQRR (mix) leads to worse ranking of gold passages than human rewrites with DE. All were deemed fluent. We found in most of these cases, CONQRR rewrites contain less context than human rewrites (56%) or additional information with a misin-

Dialogue Context	<p><i>Q</i>: What is the keto diet?</p> <p>...</p> <p><i>A</i>: The Paleolithic diet, Paleo diet, caveman diet, or stone-age diet is a modern fad diet requiring the sole or predominant eating of foods presumed to have been available to humans during the Paleolithic era.</p> <p><i>Q</i>: What do they have in common?</p>
DIALKI (mix)	What do the Paleolithic diet and the stone-age diet have in common? (Rank=78)
Human	What do paleo diet and keto diet have in common? (Rank=1)

Table 4.8: Error analysis example 2: DIALKI (mix) rewrite contains a misinterpretation of the user request, which leads to worse ranking of the gold passage than the human rewrite.

terpretation of the user request (34%). Table 4.7 and 4.8 contain examples where CONQRR (mix) rewrites have worse ranking of the gold passage than human rewrites, from our error analysis. In the two examples, the DIALKI rewrite contains less context than human rewrites or a misinterpretation of the user request.

4.4 Summary

To summarize, we introduce CONQRR to address query rewriting for conversational passage retrieval with an off-the-shelf retriever. Motivated by our analysis showing both the limitations and utility of human rewrites, which are unexplored by prior work, we adopt RL with a novel reward to train CONQRR directly towards retrieval. As shown, CONQRR is the first QR model that can be trained adaptively to any off-the-shelf retriever, and achieves state-of-the-art retrieval performance on QReCC with conversations from 3 different sources. It shows better performance with zero QR supervision when compared with strong supervised baselines trained with full QR supervision.

A direction for future work includes leveraging QR to facilitate other tasks like question answering and response generation in a full CQA system, as well as sentence rewriting in a document [Choi et al., 2021]. Future investigation is needed to explore conversations with other discourse relations like asking for clarifications besides alternating questions and answers in current CQA datasets.

Limitations

We show in § 4.3.3 (Table 4.3) that compared to directly use dialogue context without QR, a QR model has great value in robustness to topic shifts when used with an off-the-shelf retriever. However, if most

Input	Topic-Concentrated			Topic-Shifted		
	MRR	R10	R100	MRR	R10	R100
Dial Context	0.643	87.7	96.9	0.312	56.2	81.9
DIALKI (mix)	0.588	84.0	96.9	0.259	48.3	77.2
Human Rewrite	0.510	79.9	95.2	0.380	61.3	86.0

Table 4.9: Results of using the dialogue context, predicted rewrite or human rewrite as the retriever input with the *finetuned* DE as the retriever.

conversations of interest are topic-concentrated, we show that using the dialogue context itself may already work well. Although we focus on the *fixed retriever* setting in this work, we illustrate in Table 4.9, that if the downstream retriever is *allowed to be fine-tuned*, our best QR model CONQRR (mix) underperforms compared to the dialogue context in both topic-concentrated and topic-shifted scenarios, and thus the benefits of QR as an intermediate step require further justification in that setting. Nevertheless, the table still shows that human rewrites have an advantage on topic-shifted conversations over dialogue contexts. Therefore, it would be interesting for follow-up studies to investigate the design of a QR model that reaches close to the performance with human rewrites on topic-shift scenarios with a fine-tunable retriever. Then, combining the dialogue context with the rewritten query for retrieval may help further improve the overall retrieval performance.

The training time of CONQRR is longer than fine-tuning a DE retriever of a similar model size (9 vs 2 hours) because for each training step of CONQRR, CONQRR needs to do autoregressive decoding to get greedily decoded and sampled q and q_s . However, re-indexing passages after fine-tuning the retriever can be very time-consuming (about 24 hours) and memory-consuming. In addition, CONQRR can also be used for any blackbox retriever such as search engines that are infeasible to fine-tune or be replaced.

Another downside of QR is that for out-of-domain and topic-shifted scenarios, QR may still require additional labels to achieve robust performance. Although we show that CONQRR (RL) initialized with T5 can work well without QR labels on the overall QReCC test set, CONQRR (RL) does show worse robustness to out-of-domain and topic-shifted examples when compared with CONQRR (mix). Therefore, training a more robust CONQRR model may still require additional annotation efforts to collect human rewrites.

CONQRR has only been tested on the standard CQA dialogue format of alternating questions and answers. To facilitate more practical use cases with more diverse dialogue acts or discourse relations (e.g., the

agent asks a clarification question to the user), further investigation is needed.

Ethical Considerations

Our work is primarily intended to leverage query rewriting (QR) models to facilitate the task of conversational passage retrieval in an open-domain CQA system. Retrieving the most relevant passage(s) to the current user query in a conversation would help to generate a more appropriate agent response. Predicted rewrites from our QR model are mainly intended to be used as *intermediate* results (e.g., the inputs to the downstream retrieval system). They may also be useful for interpretability purposes when a final response does not make sense to the user in a full CQA system, but that introduces a potential risk of offensive text generation. In addition, to prevent the retriever from retrieving passages from unreliable resources, filtering of such passages in the corpus should be performed before any practical use.

Chapter 5

INSCIT: Information-Seeking Conversations with Mixed-Initiative Interactions

5.1 Introduction

An important advantage of conversational information-seeking systems is that users do not need to come up with a very descriptive query by themselves [Webb and Webber, 2009; Rieser and Lemon, 2009; Konstantinova and Orasan, 2013]. In realistic settings, as shown in Figure 5.1, users can start with a request that is under-specified or has no direct answer, and through conversational interactions, the agent can collaboratively guide users to refine (left) or relax their queries and proactively suggest relevant information that may partially satisfy the user's information needs (right). This collaboration requires a

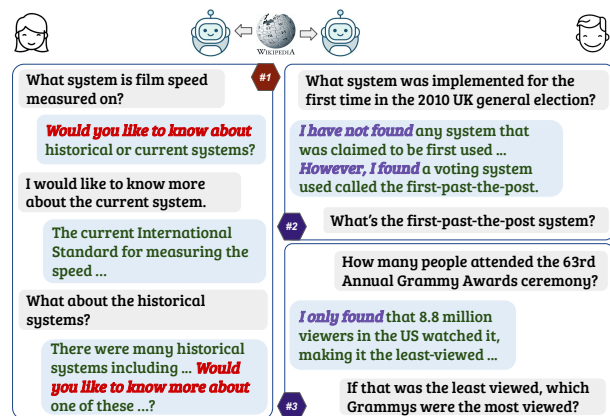


Figure 5.1: INSCIT examples show that user queries can often be under-specified and require clarification (left), or have no direct answer but where providing relevant information may fulfill users' information needs (right).

mixed-initiative dialogue, where both the user and agent can direct the flow of the conversation.

Handling such realistic user requests poses challenges to a conversational agent system. A comprehensive search can result in *multiple passages* from the knowledge source, which may provide different components of an answer (possibly incomplete) or multiple answers that surface ambiguities in the user query. Depending on the available information, the agent needs to use *different strategies*, which might involve summarizing the results, providing partial information, or trying to clarify an ambiguity. However, existing information-seeking conversation datasets rarely contain conversations where agents initiate different interaction strategies. As a result, most conversational question answering (CQA) work focuses on user-initiative interactions, where the agent simply responds to user questions with direct answers or uses `no answer` for out-of-scope queries [Choi et al., 2018; Reddy et al., 2019; Adlakha et al., 2022]. Other work studies clarification questions using artificially-created data, failing to capture natural information-seeking interactions [Saeidi et al., 2018; Feng et al., 2020; Aliannejadi et al., 2021; Guo et al., 2021]. In addition, most of them only rely on a single evidence string or passage for agent response construction.

To support research in mixed-initiative conversations, we introduce INSCIT (pronounced Insight), a dataset for **I**nformation-Seeking Conversations with mixed-initiative **I**nteractions, where agents take various strategies, such as providing direct answers (72%), raising clarifications (13%), and presenting relevant partial information (13%), to address users’ information needs. It contains 805 natural human-human conversations with 4.7K user-agent turns over diverse topics, collected through a scalable annotation pipeline and careful quality control. To simulate realistic information-seeking scenarios, users write queries with minimal restriction, and human agents decide on different strategies to respond, after searching over the knowledge source (i.e., Wikipedia) for evidence passages.

We formulate two tasks for the conversational agent system: (1) identify a set of evidence passages from Wikipedia, and (2) generate a response grounded in the evidence. Since handling queries with multiple evidence passages or no direct answer can be open-ended, we emphasize the need for human evaluation, and propose a systematic human evaluation protocol that considers diverse aspects including coherence, factual consistency and information comprehensiveness.

We present two strong baselines based on the state-of-the-art in open-domain question answering [Karpukhin et al., 2020; Izacard and Grave, 2021] and conversational knowledge identification [Wu et al., 2021d]. While

the systems achieve substantial improvements over a trivial baseline, there is still significant room for improvements, especially for scenarios requiring agent strategies other than providing a direct answer. Our analysis suggests that the key remaining challenges are improving passage identification and fusing comprehensive information from *multiple passages* by leveraging *different strategies*. We present detailed discussion and avenues for future work.

This chapter contains material that was originally published in Wu et al. [2023b]. I was the main contributor of this work, and Ryu Parish helped with setting up data collection pipeline as well as baseline models. We discussed and wrote the paper with the other collaborators throughout the project.

5.2 Task Formulations

We define two task formulations for INSCIT, namely *passage identification* and *response generation*. These two tasks mimic how an agent responds to each information-seeking user request, by first searching for relevant information over the knowledge source and then constructing the response based on the gathered information. Comparing with prior studies on open-domain information-seeking conversations [Anantha et al., 2021; Adlakha et al., 2022], the key challenges in our tasks come from identifying and fusing comprehensive information from *multiple passages* to construct responses using *different strategies*, rather than a single passage and a short answer.

At the n^{th} agent turn, both tasks have the same input: all previous utterances (i.e., dialogue context) $X = [u_1, a_1, u_2, a_2, \dots, u_n]$, the corpus of all passage candidates \mathcal{C} , and the previously used passages $\{\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_{n-1}\}$ where each $\mathcal{P}_i = \{p_i^1, p_i^2, \dots, p_i^{|\mathcal{P}_i|}\}$ is the set of passages used in the i^{th} agent turn a_i . \mathcal{C} is defined as all textual paragraphs (i.e., passages) in a full Wikipedia dump.¹

For *passage identification*, we require the model to predict a *set* of passages $\bar{\mathcal{P}}_n$ from \mathcal{C} , containing comprehensive and relevant information to the current user request u_n in the dialogue context X , which serves as evidence for the *response generation* task—generating the next agent response \bar{a}_n . This is different from the passage retrieval task where only a ranked list of relevant passages is predicted. Identifying specific knowledge to be used in the response can be important for model interpretability purposes as well as for evaluating how well a model grounds the response generation in the knowledge source. Ideally, all factual

¹We use the dump of 04/20/2022.

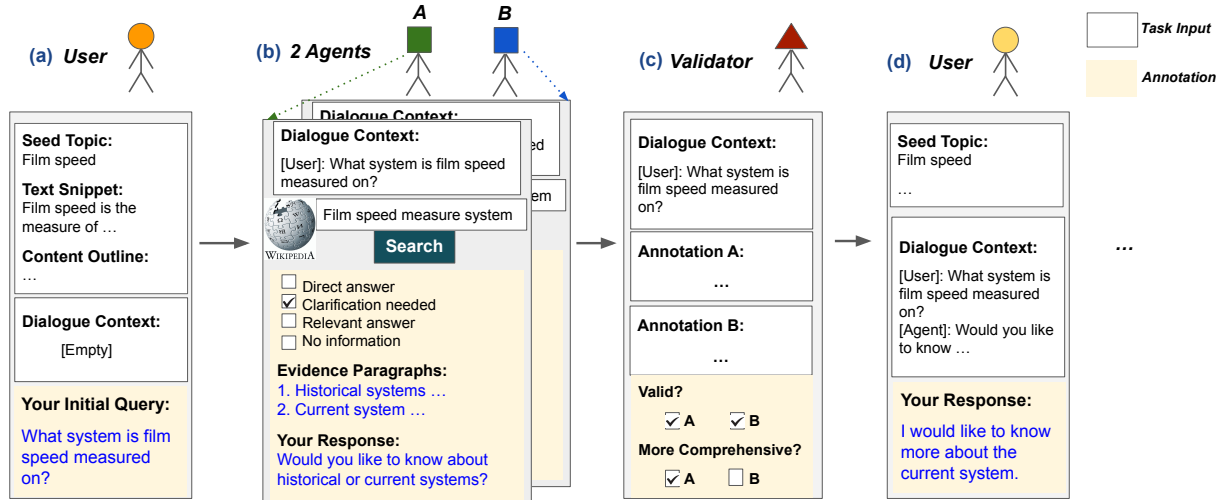


Figure 5.2: Each conversation is annotated in a series of user \rightarrow agent \rightarrow validator tasks. One worker is dedicated to each user/validator task but two workers work in parallel on the agent turn annotation (see discussion in § 5.3.1).

information contained in \bar{a}_n should be consistent with $\bar{\mathcal{P}}_n$, and every passage in $\bar{\mathcal{P}}_n$ should provide at least one unique information piece as evidence for \bar{a}_n .

In interactive dialogues, each predicted evidence $\bar{\mathcal{P}}_i$ and response \bar{a}_i are used in the dialogue context for later conversations. However, to use pre-collected dialogues with automatic evaluation metrics, the input context must be the same as that leading to the human reference response. This is also consistent with setups in previous information-seeking dialogue studies that are discussed in § 2. Therefore, the gold $\{\mathcal{P}_i\}$ and $\{a_i\}$ are used here as inputs in testing.

5.3 Data: INSCIT

INSCIT is a new information-seeking conversation dataset where the agent interprets the user intent and provides comprehensive information grounded in Wikipedia via natural human-human interactions. In this section, we present our data collection pipeline, quality control mechanisms, and analyses that show the characteristics and diversity of the user and agent turns.

5.3.1 Data Collection Pipeline

We recruit user, agent and validation workers² to create and annotate user/agent turns and validate agent annotations, respectively. Due to the asymmetric time spent by the user and the agent workers in a conversation, we design a separate annotation task for each user or agent turn, following Wen et al. [2017] to annotate each dialogue in a pipelined fashion. This framework has proved to be efficient while maintaining the conversation coherence by requiring each worker to read all previous utterances. Our data collection has IRB approval and is deemed exempt.

Figure 5.2 illustrates the data collection and annotation pipeline. Each conversation starts with an initial user turn, where the worker asks a question after reading a text snippet from a seed document. Then, two agents independently search for relevant passages in Wikipedia, provide a response, and categorize their response. Validation follows after each user-agent turn. We refer to the retrieved passages, contributed text, and validations collectively as “annotations.” The user/agent/validation process is repeated for 7 turns or until responses are found to be invalid. Details for each step follow.

Seed Document Selection To diversify conversation topics, we sample seed Wikipedia articles, used for triggering initial user requests, from 5 different topic categories—food and drink, hobby, historical events, geography and weekly top-25 pages. Additionally, we leverage the top-down tree structure of Wikipedia categories³ and sample articles at various tree depths under each of the first 4 categories. Weekly top-25 pages are from Wikipedia weekly reports of 2021.⁴ Figure 5.3 (left) shows the distribution of sampled seed documents under each category and their corresponding depths.

User Turn Here, a user worker is asked to write an *initial query* or *follow-up response* to continue the existing conversation. To trigger each conversation (Figure 5.2 (a)), the user worker is presented with the leading paragraph of a seed article, and is instructed to ask a question they are interested in but cannot find the answer from the paragraph. The article content outline containing all section titles is also provided to help with the question construction. The annotation for each following user turn (d) starts after the completion of the previous agent annotation (b) and the validation step (c), based on all previous conversation utterances.

²We use Amazon Mechanical Turk (<https://www.mturk.com/>) for data collection.

³<https://en.wikipedia.org/wiki/Wikipedia:Contents/Categories>

⁴https://en.wikipedia.org/wiki/Category:Wikipedia_Top_25_Report

Agent Turn Different from the user worker, in addition to the dialogue context, each agent worker (Figure 5.2 (b)) is given all evidence paragraphs used by each previous agent turn as additional context. Then, the worker is told to use the provided search engine⁵ to find answer(s) from Wikipedia for the current user request. They are asked to select all (up to 4) evidence paragraphs from Wikipedia, which they then use to construct their response. They are also asked to categorize their response, choosing one of four response strategies: {*direct answer* (DIRECT), *clarification* (CLAR), *relevant answer* (REL), and *no information* (NI)}. In contrast to a direct and complete answer, we consider a response as a *relevant answer* when the agent finds information that only partially satisfies the user need (e.g., relax a constraint in the request). For each agent turn, we collect two different annotations to increase reference diversity.

Validation After each user turn, we send the two agent annotations to a validator (Figure 5.2 (c)). For each agent turn annotation, the validator determines whether i) each selected evidence paragraph is properly used in the response; ii) the response is factually consistent with the evidence; iii) the response is coherent to the dialogue context; and vi) the labeled response strategy is faithfully reflected in the response. If both are valid, the validator is asked to rate which one is more comprehensive, where a tie is permitted. An agent response is considered as more comprehensive if it contains more information relevant to the user request. The more comprehensive (or the only valid) annotation⁶ is then used to continue the conversation. The annotation is terminated if both annotations are invalid, and we include the conversation up to the previous turn in our data.

5.3.2 Quality Control

Worker Qualification To recruit agent workers, we manually review > 150 submissions of a qualification task and select 24 highly qualified workers who consistently produce valid annotations during the qualification. The qualification task consists of 12 agent annotation tasks, where each dialogue context is written by the first two authors of this work. Similarly, we create different qualification tasks to select 35 qualified users and 10 validators who consistently produce reasonable user responses or validations based on our manual review.

⁵based on Google Search API from <https://developers.google.com/custom-search> and restricted to the <https://en.wikipedia.org/> domain.

⁶We randomly select one if there is a tie.

Annotation Control To discourage users from chit-chatting or raising inappropriate requests (e.g., too subjective), each agent worker can decide to either continue the conversation or flag their previous user turn as incoherent or an invalid request. The validation process ensures that only valid agent annotations are included in our final dataset. To encourage extensive search for comprehensive information, we assign a bonus to an agent worker if their annotation is labeled as equally or more comprehensive than the other worker.

We constantly monitor the annotation process and send feedback to workers. Our user and agent workers have over 99% and 96% average passing validation rate respectively. About 13% of agent annotations are marked as less comprehensive.

Worker Payment Structure We actively communicate with workers throughout the annotation process to clarify any questions they have and to give them feedback. We also check in with them early on to make sure they are satisfied with the pay and bonus structure. Most workers report that they are paid with an hourly rate of 15-20 USD, depending on their annotation speed. We pay 0.2/0.5/0.5 USD for each user/agent/validator annotation, plus a 0.1 USD bonus for each agent annotation if the worker passes validation over 80% of the time (all qualified). In addition, we assign a bonus of 0.3 USD to the agent annotation that is marked as equally comprehensive as its peer annotation by the validator, or 0.5 to those marked as more comprehensive or with multiple evidence passages found.⁷ On average, we pay over 0.9 USD to each agent annotation.

5.3.3 Data Analysis

We collect 805 conversations, which includes 4712 user-agent turns after dropping agent annotations if their evidence passages cannot be found in the post-processed Wikipedia corpus.⁸ Table 5.1 shows summary statistics of the train/dev/test subsets of INSCIT. Word token counts are based on the spaCy [Honnibal et al., 2020] tokenizer. The *test set* contains conversations triggered with seed documents from all 5 topic categories, while the *training* and *dev sets* only contain those from “food and drink”, “hobby” and “top-25”.

⁷At the beginning of our training set collection (before the collection of dev/test sets), we only assign a 0.3 USD bonus to agent annotations marked as more comprehensive. After communicating with our workers, we adjust our bonus structure, which leads to more comprehensive agent responses.

⁸We use wikiextractor to process Wikipedia articles: <https://github.com/attardi/wikiextractor>.

	Train	Dev	Test	Total
# Convs	250	86	469	805
# Turns	1443	502	2767	4712
# Turns / Conv	5.8	5.8	5.9	5.9
# References / Turn	1.8	1.6	1.6	1.7
# Tokens / User	10.6	10.5	10.7	10.6
# Tokens / Agent	35.7	44.3	45.1	41.9
# Passages / Agent	1.5	1.7	1.6	1.6

Table 5.1: Overall statistics of INSCIT.

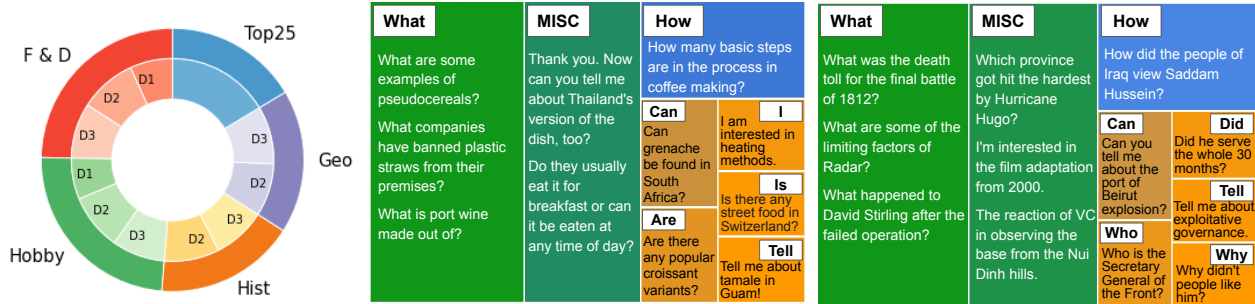


Figure 5.3: Left: seed document topic category breakdown ($D \rightarrow$ category depth). Middle and right: treemaps of top 7 (and other: MISC) first tokens in user turns from conversations under “food & drink” and “historical events” topic categories. For each figure, the size of each colored area is proportional to its percentage in the data.

In the training set, we keep all valid agent annotations as well as their comprehensiveness comparison results. In the dev and test sets, we did not include agent responses flagged as less comprehensive during validation. In addition, as discussed in § 5.3.2, we adjust the worker incentives to obtain more comprehensive responses when collecting dev/test sets, leading to the difference in the average agent turn length.

Diversity of User and Agent Turns

User Request We analyze the distribution of wh-words of user questions, and non-question user utterances (e.g., responses to clarification). The treemaps in Figure 5.3 (middle and right) show the 7 most frequent leading unigrams of user utterances in “food & drink” and “historical events” conversations respectively. “MISC” refers to utterances with less frequent leading unigrams. Each box size is proportional to its percentage in the data. As shown, most user requests are “what” and “how” questions. There are also many user turns starting with words like “can” and “tell”, most of which are responses to agent clarification questions. The user utterances are fairly long-tailed as “MISC” shares a large portion (about 30%) for both treemaps. Instead of being mostly factoid questions, open-ended user requests are well represented in INSCIT.

	DIRECT	CLAR	REL	NI
% Turns	71.5	12.7	13.1	2.7
# Tokens	43.7	33.5	46.6	10.6
# Passages	1.5	2.8	1.4	0.0

Table 5.2: Agent response strategy statistics. DIRECT, CLAR, REL, and NI indicate direct answer, clarification, no direct but relevant answer, and no information.

Clarification (CLAR)	
Too long / many answers (86%)	<p>... <Agent>: In the Battle of New Orleans, ... the Americans had 13 dead, totaling 304 dead.</p> <p><User>: Were there any long-term consequences that came as a result of the War of 1812?</p> <p><Agent>: There were many! Would you like to know what they were for <i>Bermuda, ... , or Great Britain?</i></p>
Ambiguous entity (13%)	<p><User>: Washington University is classified as what for its high research activities?</p> <p><Agent>: Do you want to know about <i>Washington University in St. Louis or in Baltimore, Maryland?</i></p>
No Direct but Relevant Answer (REI)	
Constraint relaxation / No definite answer (70%)	<p><User>: Was the Matrix franchise adapted from any other work?</p> <p><Agent>: While not explicitly adapted from another work, <i>the Matrix did draw heavily on Jean ...</i></p> <p>... <User>: Who authored the Torah?</p> <p><Agent>: <i>It was originally thought that a majority of the Torah was written by ... However, it's now thought that ... though the number of authors is up for debate.</i></p>
Relevant but side info only (29%)	<p>... <User>: What countries have an ecological footprint under 5 hectares per person?</p> <p><Agent>: <i>The world-average ecological footprint in 2013 was 2.8 global hectares per person ... But I don't have a list of countries with an ecological footprint under 5 global hectares per person.</i></p>

Table 5.3: Examples of clarification and no-direct-but-relevant-answer agent responses. *Factual information from evidence passages is italicized in agent responses.*

Agent Response Strategy Table 5.2 shows the diversity of agent response strategies in INSCIT. When no direct answer exists, agents in INSCIT can respond to the user with a *relevant answer* (see § 5.3.1). If no direct or relevant answer is found, the agent can then respond with *no information*. The average response length and number of evidence passages differ dramatically across various response strategies. Compared with direct or relevant answer cases, *clarification* responses tend to be shorter and are more likely to happen when more evidence passages are present. We also calculate that 30% *direct* or *relevant answer* agent turns have multiple evidence passages, which potentially require information summarization.

Analysis of Agent Initiatives

In this section, we present qualitative analysis to understand how different agent initiatives get triggered, with a focus on *clarification* and *relevant answer* agent responses.

Different Evidence; Different Response Strategies

Dialogue Context: ... <User>: What are some of the health consequences of oats?

<Agent> (CLAR): Oat related health concerns include oat allergies, and gluten-related disorders. Which type of health issue would you like to learn more of? (*evidence from Wikipedia page: Oat Sensitivity*)

<Agent> (DIRECT): Oat toxicity in people with gluten-related disorders depends on the oat cultivar consumed because the immunoreactivities of toxic prolamins are different among oat varieties. Furthermore, oats are frequently cross-contaminated with the other gluten-containing cereals ... (*evidence from Wikipedia page: Oat*)

Same Evidence; Different Response Strategies

Dialogue Context: ... <User>: Who is the major competitor of PepsiCo?

<Agent> (CLAR): Would you like to know PepsiCo's major competitor in the beverage market or the snack food market?

<Agent> (DIRECT): PepsiCo has two main competitors in different market sectors. The Coca-Cola Company is its largest competitor in the beverage market, and Kraft Foods is its largest competitor in the snack foods market.

Dialogue Context: ... <User>: Were the directors involved in any controversy?

<Agent> (REL): I couldn't find anything about themselves in any controversy but their film ... had a controversial storyline and themes.

<Agent> (DIRECT): The Wachowskis' V for Vendetta was considered to have controversial storyline and themes, but have been both criticized and praised by sociopolitical groups.

Table 5.4: Examples of two agent reference responses with different response strategies.

... <User>: What kinds of regional varieties are there?

<Agent>: Would you like to know about East Asia, Southeast Asia, South Asia, or Europe?

<User>: Tell me about East Asia.

<Agent>: Sorry, but each country is detailed as well, do you want to know more about congee in China, Japan, Korea or Taiwan?

Table 5.5: An example of consecutive clarifications.

Fine-Grained Categorization We randomly sample and analyze 100 clarification and relevant answer responses respectively. Table 5.3 (upper half) shows that in most cases, the agent raises a clarification when they find a long answer or too many answers (86%) or notice an ambiguous entity in the user request (13%). In 70% of relevant answer cases (bottom half of Table 5.3), the agent relaxes some constraint in the user request or provides evidence that no definite answer can be found. In 29% of these cases, they simply provide some relevant but side information only. We also observe that in rare cases (1%), the agent points out some mistake (e.g., a false assumption) in the user request.

Clarification Occurrences We next look at contexts where agents are more likely to ask for clarification in a conversation. Clarification questions occur more frequently at the very beginning (ex. 2, Table 5.3), rather than later in a conversation (18.8% vs. 11.5%). If a clarification is raised in the previous agent turn, the chance of a subsequent clarification (see Table 5.5) is 7.6%, compared to 12.2% if the previous turn is not a clarification (ex. 1, Table 5.3).

Response Strategy Selection In 23% of examples with 2 agent annotations marked as equally comprehensive by validators, workers take *different response strategies given the same dialogue context*. Of this set, 82% have different evidence passages labeled by the two workers, potentially due to the open-endedness of user queries in INSCIT and the large knowledge source. In addition, as suggested by our analyses in § 5.5.2, it is more likely that agents will choose different evidence passages when there is no direct answer to the question. As illustrated in the first example in Table 5.4, the different evidence passages often trigger different agent response strategies. The second and third examples show that even if two agents find the same evidence set, deciding whether it indicates an under-specified user request, a direct or only a relevant answer can be subjective.

5.4 Experiment Setup

5.4.1 Systems

We build two systems for each of the tasks formulated in § 5.2. Both systems build on retriever-reader models, inspired by recent advances in open-domain single-turn or conversational question answering [Karpukhin et al., 2020; Izacard and Grave, 2021; Adlakha et al., 2022]. Here, the main function of the retriever is to gather a ranked set of top-k candidate evidence passages from the entire Wikipedia to facilitate passage identification and response generation for the later reader model. We first describe the retriever models, and then introduce the two reader models that perform the two main tasks based on retrieval results.

Retriever Models

We experiment with two retrievers: BM25 and DPR. **BM25** [Robertson and Zaragoza, 2009] uses sparse bag-of-word representations for ranking passages with regard to each query. We use Pyserini [Yang et al., 2017] in our experiments. **DPR** [Karpukhin et al., 2020] is a BERT-based [Devlin et al., 2019] dual encoder model, that produces learned dense representations for queries and passages, and measures the relevance using the dot product similarity in the vector space. We finetune DPR on INSCIT. As the training set is small in INSCIT, we initialize it with a downloadable checkpoint⁹ that is pre-trained on a much larger (>

⁹<https://github.com/McGill-NLP/topiocqa>

30×) open-domain conversational question answering dataset, TopioCQA [Adlakha et al., 2022].

Reader Models

Our two readers are based on state-of-the-art models in open-domain question answering and conversational knowledge identification—Fusion-in-Decoder [Izacard and Grave, 2021] and DIALKI (Chapter 3).

Fusion-in-Decoder (FiD) FiD is a generative reader model. It first encodes all retrieved passages with a given query, and then decodes the task output (e.g., an answer string) by attending over all encoded passages. To adapt FiD to our tasks, we prepend a passage identifier (ID) to each of the top- k retrieved passages (here, $k = 50$, following Adlakha et al. [2022]) and separately concatenate each passage with the dialogue context for encoding. Given the 50 encoded contextualized passage vectors, the decoder generates a sequence of evidence passage IDs (*passage identification*), followed by the final response (*response generation*). After the first turn, the encoded passage vectors associated with $\{\mathcal{P}_i, \dots, \mathcal{P}_{n-1}\}$ are concatenated with the top- k retrieved passages, limiting k to give a total of 50. In training, we use the same hyperparameters as in Adlakha et al. [2022], with the batch size adjusted for the memory constraint and training steps adjusted to have the same epochs.

DIALKI + FiD The second reader adopts a pipelined approach to perform the two tasks. It first uses DIALKI (Chapter 3) to select evidence passages and then feeds the identified passages into FiD to generate the agent response. DIALKI predicts a passage score for each input passage (i.e., each top- k retrieved passage). To adapt it for passage identification, we simply keep evidence passages (up to 4, as in data collection) with ranking scores higher than γ for *multiple passage prediction*, where γ is tuned on the dev set. We apply the same method to incorporate previously used evidence passages into DIALKI as in the first reader model. We set the number of input passages of DIALKI to be 50 and keep other original hyperparameters. Parameters in FiD are the same as the first reader model, except that the number of input passages is 4 in the DIALKI+FiD system.

Trivial Baseline: Last Turn We report performance of a simple baseline: use the most recent agent turn in the dialogue context and associated evidence ($\bar{\mathcal{P}}_n = \mathcal{P}_{n-1}$; $\bar{a}_n = a_{n-1}$). For first-turn instances, we use

the most frequent evidence passage and agent response seen in the training set as the prediction. We also tried using a random previous turn as the prediction, which gives lower scores than using the last turn.

Human We collect one additional annotation for each agent turn in the test set and evaluate it as the human performance. These additional annotations are annotated by the same agent workers we select in § 5.3.2. Note that these additional prediction data do not go through the same validation step as those that are used as references.

5.4.2 Evaluation

Below, we describe automatic metrics and a new human evaluation protocol for the passage identification (PI) and response generation (RG) tasks in § 5.2.

Passage Identification INSCIT allows for multiple evidence passages, so we measure the model performance by computing the F1 score (PI-F1), comparing the set of predicted evidence passages $\bar{\mathcal{P}}_n$ to the set of reference passages \mathcal{P}_n . For turns where there are two valid reference annotations, we use the maximum F1 score between the two.

Response Generation For a generated agent response \bar{a} , we calculate the SACREBLEU score [Post, 2018] (BLEU in tables) and token-level F1 (RG-F1) scores against the reference response, following previous studies [Feng et al., 2020; Adlakha et al., 2022]. Again, when there are two valid annotations, we use the maximum.

Human Evaluation As the two tasks are dependent on each other, decoupled automatic evaluations may not capture aspects like factual consistency between predicted passages and the response. Moreover, handling queries with multiple evidence passages or no direct answer can be open-ended.

Therefore, we design a human evaluation protocol to evaluate the model performance on both tasks. Specifically, we focus on the evaluation of 4 dimensions: 1) *evidence passage utility* (EU): how many predicted evidence passages are used in the generated response; 2) *factual consistency* (FC) between the predicted response and evidence; 3) response *coherence* (CO) with the dialogue context; and 4) response

comprehensiveness (COMP): how much information, that is both relevant to the user request and factually consistent with the predicted evidence, is contained in the response. While most prior work on information-seeking dialogues only relies on automatic metric scores [Choi et al., 2018; Anantha et al., 2021; Adlakha et al., 2022], a few studies collect human ratings on dimensions like response “coherence” and “informativeness” [Gao et al., 2022; Feng et al., 2022]. However, as they do not require models to predict evidence, the factual consistency between the response and the knowledge source cannot be evaluated [Nakano et al., 2021].

We provide outputs for both tasks of our two systems and “Human” to a human judge. We ask them to rate the first 3 dimensions for each system output on a 4- or 5-point Likert scale¹⁰ and then rank the system responses in terms of *response comprehensiveness* (ties are permitted). We have 3 raters for each agent turn and take the average rating score or rank place on each dimension for each system. Since human evaluation can be time-consuming and costly, we run it on a sampled test subset with 50 conversations (290 turns) and encourage future studies to report on the same subset.

The inter-rater agreement measured as Krippendorff’s alpha is 0.66, 0.64, 0.42 and 0.37 for EU, FC, CO and COMP, respectively, which can be interpreted as good or moderate agreements. We observe two main types of coherence disagreements: 1) some workers are more strict and indicate one response as more preferred due to minor differences (e.g., a connecting word), or 2) both responses are incoherent, but in very different ways (e.g., have very different content). Similarly, most comprehensiveness disagreements involve either: 1) two responses that are similar except that one includes additional side information, or 2) two responses that provide different answers but both are good.

5.5 Experiment Results

5.5.1 Quantitative Results

Table 5.6 shows the overall automatic evaluation results of all systems for our main tasks (PI and RG) on the *dev* set. The simple baseline performs very poorly. Using retrieval results from DPR (vs. BM25) leads to the best overall performance for both tasks. For both BM25 and DPR retrievers, DIALKI + FiD

¹⁰The 4-point scale is used only for coherence to discourage *neutral* ratings. We report all scores normalized to a 1-5 scale.

Retriever	Reader	PI-F1	BLEU	RG-F1
Last Turn	Last Turn	10.5	4.2	14.1
BM25	FiD	14.1	9.4	22.5
	DIALKI + FiD	17.0	13.8	24.8
DPR	FiD	17.1	8.8	21.6
	DIALKI + FiD	21.5	16.6	26.6

Table 5.6: Automatic scores on the **dev** set.

Retriever	Reader	Automatic			Human		
		PI-F1	BLEU	RG-F1	EU	FC	CO
DPR	FiD	17.5	9.6	22.2	2.35	2.52	3.76
	DIALKI + FiD	23.7	16.0	27.8	4.33	4.74	3.77
-	Human	52.5	33.8	43.5	4.76	4.77	4.85

Table 5.7: Automatic scores on the **test** set, and human scores on 50 *sampled* test conversations (290 turns) for dimensions rated with Likert scales: evidence utility (EU), factual consistency (FC) and coherence (CO).

achieves better performance than FiD in all metrics. A possible reason could be that the smaller number of context vectors used with DIALKI+FiD is better suited to learning from limited data than the end-to-end FiD approach. DIALKI leverages previous evidence passages in passage identification, so its following FiD response generation has only 4 context vectors (vs. 50 for FiD). This hypothesis is also supported by a separate observation during our experiment that incorporating previously used evidence hurts the RG performance slightly for FiD but for DIALKI+FiD it helps (roughly 1 point decrease vs. increase in scores, respectively, with DPR).

Table 5.7 shows both automatic and human evaluation results on the *test* set for FiD and DIALKI+FiD with the DPR retriever, confirming the dev set findings. Experiments with BM25 also confirm dev set trends. Figure 5.4 presents comparative human evaluation results. DIALKI+FiD greatly outperforms FiD except in coherence where scores are similar. DIALKI+FiD substantially underperforms the human "system" in both automatic and human evaluation scores, except for factual consistency where the difference is small. This could indicate that, although DIALKI+FiD generates responses consistent with the predicted evidence, it identifies less relevant passages which lead to less coherent and less informative responses.

The reason for *imperfect human performance* on passage identification, shown in Table 5.7, is two-fold. Due to the open-endedness of information-seeking queries in INSCIT and the large search space over Wikipedia, annotators may find different (but both valid) sets of evidence passages. In addition, annotations

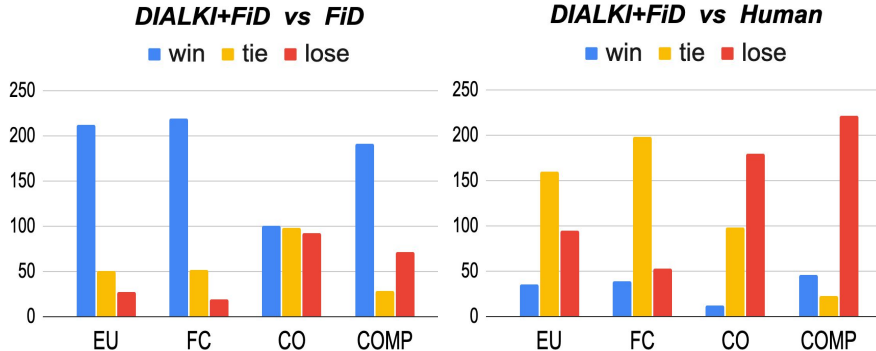


Figure 5.4: Human evaluation on system comparison for all dimensions: evidence utility (EU), factual consistency (FC), coherence (CO) and response comprehensiveness (COMP). Win/lose refers to DIALKI+FiD.

Retriever	Dev		Test	
	HIT@20	HIT@50	HIT@20	HIT@50
BM25	35.3	48.0	35.6	48.1
DPR (FT only)	62.5	70.1	51.3	60.8
DPR (PT only)	66.4	76.3	68.4	77.5
DPR	71.1	79.8	69.9	77.5

Table 5.8: *Passage retrieval* results. PT and FT refer to pretraining on TopioCQA and finetuning on INSCIT.

corresponding to the Human “system” do not go through the validation process, so they could have errors or be less comprehensive.

5.5.2 Analysis

Passage Retrieval Table 5.8 reports the performance for passage retrieval in HIT@k scores, following Karpukhin et al. [2020]; Adlakha et al. [2022]. HIT@k is calculated as $\mathbb{1}[|\mathcal{R}_K \cap \mathcal{P}| > 0]$, where \mathcal{R}_K denotes the top K retrieved passages and \mathcal{P} denotes the union of the two reference passages sets (or a single reference set if only one is valid). We evaluate both BM25 and DPR models used in our main experiments, as well as two DPR ablations: with pretraining (PT) on TopioCQA or finetuning (FT) on INSCIT only. BM25 underperforms DPR models significantly, which explains the main task performance differences between BM25 and DPR in Table 5.6. DPR with PT alone is more effective than FT only, which can be explained by the much larger training data in TopioCQA. The best retrieval results are achieved with PT and FT combined. We do not leverage TopioCQA for pretraining on our two main tasks, because 1) it does not come with the passage identification task and only has short answers or no answer as their agent

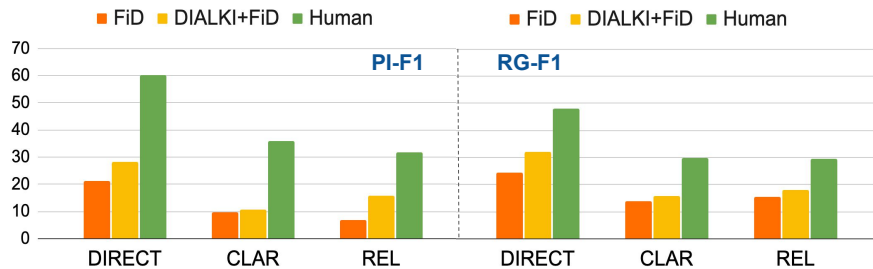


Figure 5.5: PI-F1 and RG-F1 scores by reference response strategy (direct answer, clarification, relevant answer) on the one-strategy test subset, excluding instances where two references differed in strategy.

responses; 2) we observe poor zero-shot response generation performance on INSCIT for FiD trained on TopioCQA.

Passage Identification & Response Generation Performance Breakdown Figure 5.5 shows the system and task performance breakdown by *reference response strategy* (direct answer, clarification and relevant answer) for the test set, excluding examples where two annotations differed in the response strategy category (16%). DPR is used for retrieval. Only RG-F1 is shown for response generation; trends for BLEU are similar. For all response types, DIALKI+FiD is similar or outperforms FiD, but significantly underperforms humans. For both systems and humans, the non-direct-answer responses have lower automatic scores. The lower PI-F1 scores for humans suggest that the retrieval task is more difficult (with more variety in evidence) when a simple direct answer is not available. Lower automatic response generation scores may be explained by lower retrieval scores (less reliable evidence), larger number of passages, and/or challenges in learning non-direct-answer response strategies. Note that for both systems, the largest percentage gap with respect to human scores is for clarifications.

Response Generation Results with Human Evidence Passages To explore the above hypotheses, we generated responses using the DIALKI+FiD response generator with passages selected in the “Human” annotation of the test data. The resulting responses had 26.5 and 37.4 for BLEU and RG-F1 scores, respectively, compared to 16.0 and 27.8 when using DIALKI passages. We sample and analyze 20 examples each of single and multiple “Human” evidence passages. Given multiple evidence passages, most DIALKI+FiD responses either do not use all passages or introduce incorrect facts. With one passage, responses are consistent with the evidence but not always as comprehensive as for humans. In the 20 examples with multiple

Model Input	Dev		Test	
	BLEU	RG-F1	BLEU	RG-F1
DC+OEP+RT (Oracle)	32.6	48.7	31.6	47.4
DC+OEP+RT (Predicted)	32.6	46.3	31.7	45.4
DC+OEP	32.0	45.3	30.6	44.3

Table 5.9: Automatic RG scores for FiD with inputs: dialogue context (DC), oracle evidence passages (OEP), and different (oracle/predicted/no) response types (RT).

passages, DIALKI+FiD asks one clarification, whereas humans ask nine.

Impact of Response Type Prediction for Response Generation As explained in § 5.3.3, the agent response type depends on the selected evidence passages. To analyze how incorporating response types can help with response generation, we conduct a controlled experiment to generate agent responses with the dialogue context and oracle evidence passages as the input to FiD, and compare the performance when no/oracle/predicted response type is given. For examples that have two labels with different sets of evidence passages, we split them into two separate instances. To predict the response type, we use a sequence classification model based on BERT-base [Devlin et al., 2019], given the dialogue context and oracle evidence passages. To provide the oracle or predicted response type as the response generation model input, we simply append a formatted string—`response type: {response_type_name}`¹¹—at the end of the dialogue context, when feeding it to FiD.

The response type classification model gives an overall accuracy of 0.75, compared to 0.73 when predicting everything as “direct answer.” Table 5.9 shows that adding either oracle or predicted response types improves BLEU and RG-F1 scores, compared with no response type being used, with greater gains in RG-F1 for oracle response type. We observe consistent performance gains on examples with either “direct answer”, “clarification” or “no information” oracle response types, but not for the “relevant answer” response type.

5.6 Ethical Considerations for Dataset Collection

Our work is primarily intended to encourage future work in information-seeking conversation *factually grounded* in given knowledge sources. Our knowledge sources come from Wikipedia articles, where the

¹¹Candidate response type names are “direct answer,” “clarification,” “no answer but relevant information” and “no answer and no information.”

content follows principles emphasizing on a neutral point of view and reliable sources. Before and during the data collection, we carefully guide our user workers not to ask subjective or opinion-driven questions, and our agent workers not to include any content without evidence from the knowledge sources in their conversational responses. Therefore, all contents exposed to our workers during data collection should contain minimal risk to the workers. Our data collection has IRB approval from University of Washington and is deemed exempt. We also actively communicated with the workers to address any concern they had and we usually replied back within an hour during the whole data collection process. This communication also helped us to make sure that our workers were compensated fairly. As explained in § 5.3.2, most of our workers report that they are paid with an hourly rate of 15-20 USD.

5.7 Summary

In summary, we introduce INSCIT, a new open-domain information-seeking conversational dataset grounded in Wikipedia, with mixed-initiative user-agent interactions. INSCIT supports two tasks (passage identification and response generation), for which we present results of two strong baselines, with best results obtained with the pipelined DIALKI+FiD system. We also introduce a new human evaluation protocol.

Both models significantly underperform humans in both tasks in all metrics. The relative performance gap is greatest for scenarios that require the agent to provide a non-direct answer. We find that passage identification significantly impacts response generation (particularly coherence) by providing relevant grounding knowledge. Thus, improving methods for selecting relevant passages is critical for future work. A key challenge that remains in response generation is how to fuse and present comprehensive information from multiple passages, which we address in the next chapter.

Chapter 6

Fine-Grained Human Feedback Gives Better Rewards for Language Model Training

6.1 Introduction

State-of-the-art AI is built on pre-trained language models that are then trained through interaction with humans [Ouyang et al., 2022; OpenAI, 2023; Fu et al., 2023], with a combination of supervised learning and reinforcement learning. Incorporating human feedback into the process of language model (LM) training has been shown as effective in reducing false, toxic and other undesired model generation outputs [Ouyang et al., 2022; Bai et al., 2022b,a; Ramamurthy et al., 2023; Ganguli et al., 2023]. Many of these studies adopt reinforcement learning from human feedback (RLHF) [Ziegler et al., 2019], a framework that converts human feedback into an effective LM training signal to reach these goals. Specifically, humans are presented with two or more outputs and asked to select one or rank them, and this signal is then used to train a reward model, which computes a single scalar reward for each LM-generated sequence. The LM is then trained with RL to optimize the reward it receives (from the reward model).

Such a reward provides a relatively sparse training signal, especially for tasks that require the generation of long-form text—making RLHF in such domains unreliable [Ramamurthy et al., 2023]. Furthermore, previous research into automated evaluation of generated text [Dou et al., 2022; Krishna et al., 2023; Gao

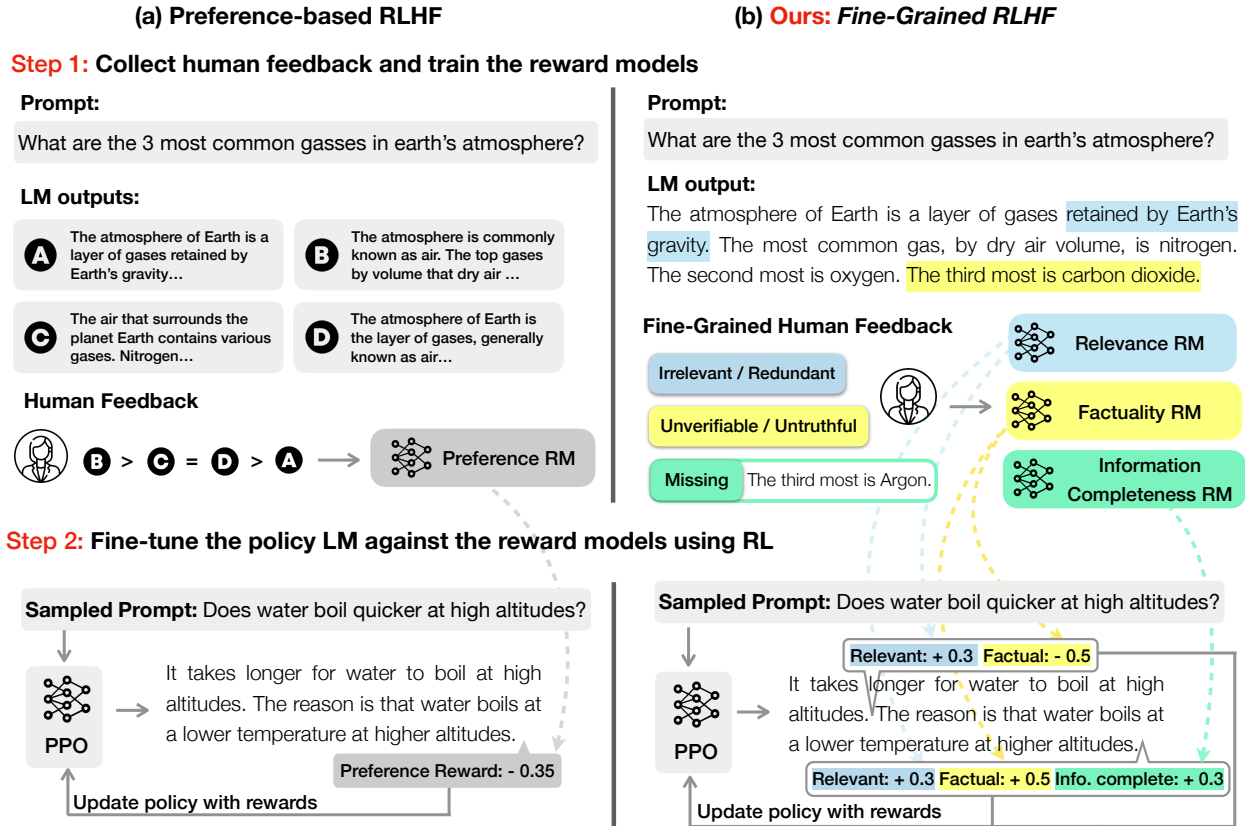


Figure 6.1: Comparison of (a) RL with human preference and (b) our FINE-GRAINED RLHF on long-form QA. Different from (a), which collects human preferences on the overall quality of LM outputs, we ask annotators to mark which part of an output contains what type(s) of errors. We train a fine-grained reward model for each type of error and optimize LM against these reward models. In this example, we provide a relevance reward and a factuality reward after each sentence is generated. There is also a holistic information completeness reward after the whole text is generated.

et al., 2023; Yue et al., 2023; Min et al., 2023; Xu et al., 2023b] shows that it can be challenging for human annotators to reliably compare the overall quality of two or more model outputs when the outputs contain a mixture of diverse undesired behaviors. They demonstrate how categorizing and localizing model errors (i.e., fine-grained evaluation) provides explicit insights about which part of the model output has what type of problem. We thus ask the question: how can we improve rewards for LM training via RLHF by using more *fine-grained human feedback*?

In this chapter, we propose that humans give fine-grained feedback to the LM, associating *categories* of undesired behavior (e.g., false or irrelevant generations) and a text span at a *density* (e.g., sentence or sub-sentence-level). To enable LMs to learn from such fine-grained feedback, we introduce the FINE-GRAINED

RLHF framework. As shown in Figure 6.1, we first use collected human feedback to train fine-grained reward models such that each of them focuses on one *category* and provides rewards at the *density* associated with that category. We then integrate these reward models into Proximal Policy Optimization (PPO) [Schulman et al., 2017], a commonly used RL algorithm for training LMs with preference-based human feedback (§6.2).

We conduct experiments on two language generation tasks—detoxification [Gehman et al., 2020] (§6.3) and long-form question answering (QA) [Stelmakh et al., 2022] (§6.4). For detoxification, toxicity is the only error category and we explore learning with a dense reward. We adopt PERSPECTIVE [per], a widely used language toxicity detection model trained on millions of human annotations, as our reward model. We use it to calculate a fine-grained reward after the generation of every sentence. Our experimental results show the efficacy and data efficiency of training models with dense reward compared to a holistic sequence-level reward, supported by multiple automatic evaluation results.

With experiments on long-form QA, we aim to examine training models with fine-grained rewards at the two granularity dimensions (density and error category), for which we construct a long-form QA dataset, QA-FEEDBACK, along with our collected human feedback. We carefully develop a pipeline to collect fine-grained human feedback on three error categories at different density levels: i) irrelevance, repetition, or incoherence (sub-sentence), ii) incorrect or unverifiable facts (sentence), and iii) incomplete information (whole sequence; see Figure 6.1). Our experimental results show improved results in each error category by learning with such fine-grained feedback, supported by both automatic and human evaluation results. In a scenario with multiple reward models representing different error types, we also show FINE-GRAINED RLHF allows us to combine reward models with different weights and thus control the model training process towards a customized combination of desired behaviors.

This chapter contains material that was originally published in Wu et al. [2023a]. Yushi Hu and I were the main contributors of this work. I initiated the project idea and then we made equal contributions in carrying out the project. I was mainly responsible for data collection and reward modeling, and Yushi was mainly contributing to RL training. We discussed and wrote the paper with the other collaborators throughout the project.

6.2 FINE-GRAINED RLHF

We introduce FINE-GRAINED RLHF, a framework that enables us to train fine-grained reward functions for generation outputs across different feedback types. We first define the RL environment and learning algorithm. Then we define the fine-grained reward models and describe how to incorporate the fine-grained reward model(s) into an RL algorithm, in contrast to previous RLHF studies that only consider a single reward.

Environment: language generation as a MDP. We focus on language generation tasks. For each task, we are given a set of task input prompts $D = \{x^n\}_{n=1}^N$. We follow Ramamurthy et al. [2023] to define language generation as a Markov Decision Process (MDP) $\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, P, \gamma, T_{max} \rangle$ with a finite vocabulary \mathcal{V} . Each MDP episode starts with a sampled prompt $x = (x_1, x_2, \dots, x_l)$ with $x_i \in \mathcal{V}$, and ends when the current time step exceeds T_{max} or an end of sequence token is generated. \mathcal{S} is the state space and $s_0 = (x_1, x_2, \dots, x_l) \in \mathcal{S}$ is the initial state. An action in the environment $a_t \in \mathcal{A}$ is a generated token (by the policy LM model P_θ) at time t from \mathcal{V} (a_0 is the begin sequence token). The transition function $P : \mathcal{S} \times \mathcal{A} \rightarrow \Delta\mathcal{S}$ appends a_t at the end of the state $s_t = (x_1, x_2, \dots, x_l, a_0, a_1, \dots, a_{t-1})$. This process continues until the end time step $T \leq T_{max}$ is reached, which gives a generated sequence $y = (a_1, \dots, a_T)$. A reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, which comes from the reward model(s) in FINE-GRAINED RLHF, provides dense rewards before and when T is reached. P_θ can be initialized with a pre-trained language model, and optionally also with supervised fine-tuning on task-specific demonstrations. The reward function is defined separately for each task.

Learning algorithm: proximal policy optimization (PPO). PPO [Schulman et al., 2017] is an actor-critic RL algorithm that is widely used in previous RLHF work to optimize the policy model against a reward model of human feedback. It uses a value model $V_\psi(s_t)$ to estimate the value of state s_t , and optimizes the policy model with a PPO clipped surrogate training objective. The advantage A_t at timestep t is estimated by a generalized advantage estimation function [Schulman et al., 2016]: $A_t = \sum_{t'=t}^T (\gamma\lambda)^{t'-t} (r_{t'} + \gamma V_\psi(s_{t'+1}) - V_\psi(s_{t'}))$, with γ as a hyperparameter and λ as the discounting factor for rewards. r_t is the reward assigned to a_t , which in our case is acquired using one or multiple learned reward models. The value model $V_\psi(s_t)$ is optimized with an expected squared-error loss with the value target as $V^{\text{targ}}(s_t) = \sum_{t'=t}^{T-1} \gamma^{t'-t} r_{t'} + \gamma^{T-t} V_{\psi_{\text{old}}}(s_T)$, where $V_{\psi_{\text{old}}}$ is the lagging value model. Finally, PPO is trained to optimize

both policy (P_θ) and value (V_ψ) models with their respective objectives. No reward model is being optimized during PPO training. See Appendix A.1 for more details.

Fine-grained reward models. Previous RLHF work adopts a holistic reward model R_ϕ that maps input prompt x and generated output y to a single scalar reward representing its overall quality (Figure 6.1(a)). This single scalar reward is only assigned to the final token in the generated sequence, a_T . Formally, $r_t = R_\phi(x, y)$ if $t = T$ and 0 otherwise.

In contrast, we consider a reward function that is derived from one or multiple *fine-grained* reward models that (1) provide rewards densely (i.e., for subsequences of the generated output), and (2) compute rewards on distinct categories of undesired behaviors (e.g., false or repetitive generation), where each category is associated with an individual reward model.

For a fine-grained reward model R_{ϕ_k} that gives feedback on error category C_k , we first segment y into L_k segments $(y_1^k, y_2^k, \dots, y_{L_k}^k)$ corresponding to the density (e.g., sentence-level) of R_{ϕ_k} , where each segment y_j^k ends at timestep T_j^k . R_{ϕ_k} outputs a reward $R_{\phi_k}(x, y, j)$ for each segment y_j^k given x and y as the input, which is assigned to the final token in y_j^k . Additionally, to ensure the fluency of generated outputs, we follow Wu et al. [2021a] and add an approximate KL divergence penalty to each token a_t with a weight β , that is not backpropagated through during training. Formally, assuming that we have K fine-grained reward models that represent different error categories, we will have a combined reward function for each token a_t as:

$$r_t = \sum_{k=1}^K \sum_{j=1}^{L_k} \left(\mathbb{1}(t = T_j^k) w_k R_{\phi_k}(x, y, j) \right) - \beta \log \frac{P_\theta(a_t | s_t)}{P_{\theta_{\text{init}}}(a_t | s_t)} \quad (6.1)$$

where $w_k \in \mathbb{R}$ is a weight assigned to reward model R_{ϕ_k} . Then we follow the same PPO training algorithm to optimize the policy model. We discuss details of how we define and train fine-grained reward models for the detoxification and long-form QA task in our experiments in § 6.3 and § 6.4 respectively.

6.3 Task 1: Detoxification

The task of detoxification aims to reduce the toxicity in the model generation y when given a prompt x . Toxicity is the only undesired behavior in this task, and we aim to explore learning with a dense reward in comparison to a *single* holistic reward. We conduct our experiments on REALTOXICITYPROMPTS, a dataset of 100K sentence-level prompts derived from the web that are known to easily elicit problematic generations

in GPT-2 [Radford et al., 2019]. Using a dense sentence-level fine-grained reward, we demonstrate that **our fine-grained reward exhibits greater sample efficiency compared to a holistic reward**, achieving lower toxicity with fewer training steps while maintaining better fluency (§6.3.1).

Holistic reward for (non-)Toxicity. We use the PERSPECTIVE API [per] as our reward model, which is widely used for language toxicity detection and is trained with millions of examples gathered from several online platforms and annotated by human annotators for toxicity. That means we use an off-policy reward model that is not trained on outputs from $P_{\theta_{init}}$. The API outputs a score between 0 (non-toxic) and 1 (toxic). Given the entire model output y , the holistic reward for RL is $1 - \text{PERSPECTIVE}(y)$.

Sentence-level (fine-grained) reward for (non-)Toxicity. To calculate the *fine-grained reward*, we query the API after the model generates each sentence instead of generating the full sequence. For each generated sentence y_j , we assign $R(x, y, j) = \text{PERSPECTIVE}([y_1, \dots, y_{j-1}]) - \text{PERSPECTIVE}([y_1, \dots, y_j])$ as the sentence reward (i.e., how much toxicity is changed from generating y_j). Since there is only one error category, we omit the category indicator ϕ_k superscript k , using y_j to denote the j^{th} segment (e.g., sentence) in y .

6.3.1 Experiments

Implementation details. We follow previous work [Krause et al., 2021; Liu et al., 2021] and use GPT-2 large model as the initial policy model $P_{\theta_{init}}$. During both the exploration stage in RL training and inference, we use nucleus sampling decoding with $p = 0.9$ and temperature = 1.0. The generation length limit is set to 48. The value model used during RL training is initialized with GPT-2-base due to GPU memory constraint. We report RL training parameters in Appendix A.1. All scores are averaged over 3 independent runs.

Compared systems and evaluation. We report the performance of **FINE-GRAINED RLHF**, RLHF with holistic reward (**Hol. RLHF**), and the state-of-the-art controlled generation approaches **GeDi** [Krause et al., 2021] and **DEXPERS** [Liu et al., 2021]. We follow previous work [Krause et al., 2021; Liu et al., 2021] to report the toxicity score calculated on each full generation sequence from the PERPLEXITY API, as well as other commonly used metrics for REALTOXICITYPROMPTS, including n-gram diversity and GPT-2 XL perplexity (PPL) as a proxy for fluency. The lower the perplexity, the more fluent the generated text. The toxicity score is reported as the *maximum* score among 4 sampled model outputs, averaged over all test input prompts. Other metrics are reported as the *average* score of the same 4 samples.

	Toxicity	Fluency	Diversity	
	avg max (\downarrow)	PPL (\downarrow)	dist-2 (\uparrow)	dist-3 (\uparrow)
GPT-2	0.192	9.58	0.947	0.931
Controlled Generation				
GeDi	0.154	24.78	0.938	0.938
DEXPERTS	0.136	22.83	0.932	0.922
Hol. RLHF	0.130	11.75	0.943	0.926
F.G. RLHF	0.081	9.77	0.949	0.932

Table 6.1: Results on the REALTOXICITYPROMPTS test set.

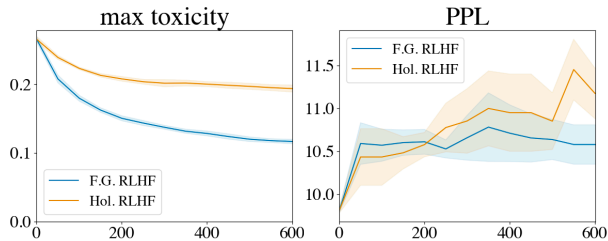


Figure 6.2: Curves of toxicity and perplexity on the dev set vs. training steps.

Main results. Table 6.1 shows the experimental results on the REALTOXICITYPROMPTS test set. FINE-GRAINED RLHF with sentence-level fine-grained reward attains the lowest toxicity and perplexity among all methods, while maintaining a similar level of diversity.

Sample efficiency analysis. Figure 6.2 shows the max toxicity and average perplexity on the development set during training. FINE-GRAINED RLHF has the toxicity drop much faster while keeping a low-level perplexity. This shows that learning from a denser fine-grained reward is more sample efficient than a holistic reward. One explanation is that the fine-grained reward locates where the toxic content is, which is a stronger training signal compared with a scalar reward for the whole text. The cost is that we have to query the reward model more times per example.

6.4 Task 2: Long-Form Question Answering (QA)

Long-form QA requires an LM to generate a textual response to a question with a comprehensive answer and explanation. To examine learning with fine-grained rewards at the two granularity dimensions (error category and density), we collect QA-FEEDBACK (§6.4.1), a long-form QA dataset annotated with human feedback on LM-generated responses. We define three error categories at different density levels and train a reward model for each (§6.4.2). We describe the experimental setup in §6.4.3. Both human and automatic evaluation show that FINE-GRAINED RLHF outperforms preference-based RLHF and supervised fine-tuning models on all error categories (§6.4.4). We then show that adjusting the weights of fine-grained reward models during RL training leads to distinct behaviors in LM generation, allowing us to customize the LM for users with different needs (§6.4.5). Finally, we conduct an in-depth analysis of the fine-grained reward models, revealing that they compete against each other, and provide an analysis of their impact on

the resulting policy model.

6.4.1 QA-FEEDBACK: Long Form QA with Human Feedback

QA-FEEDBACK is based on ASQA [Stelmakh et al., 2022], a dataset that focuses on answering ambiguous factoid questions [Min et al., 2020] in an open-domain setting. We use their provided oracle knowledge contexts to reformulate the task into a reading comprehension setting: given the input x that contains a question q and a set of knowledge passages $P = \{p_1, \dots, p_{|P|}\}$, generate a long-form response y . On average, there are roughly 65 words in each gold response. To construct P for each input x , we use the oracle knowledge contexts provided by ASQA for each x , that are text snippets from the passage corpus. We use BM25¹ to map each knowledge context (text snippet) to the closest passage from the passage corpus and use the resulting passages as P . Since ASQA does not release the test set, we create our own train/development/test data split from the original train and development sets. Our train and dev examples come from the original ASQA train set and our test examples are the original ASQA dev examples. Overall, QA-FEEDBACK has 3,853 training, 500 development, and 948 test examples.

Initial policy. Before collecting human feedback, we follow Ramamurthy et al. [2023] to initialize the policy model with supervised fine-tuning on a small set of examples. Specifically, we use 1K training examples to supervise fine-tuning of T5-large (the original baseline for ASQA) [Raffel et al., 2020] to get $P_{\theta_{init}}$. We name this initial policy model **SFT**. We then sample outputs from SFT for the remaining training and development examples and collect *fine-grained* human feedback in three error categories— C_1 : **irrelevance, repetition, or incoherence**; C_2 : **incorrect or unverifiable facts** based on knowledge passages; and C_3 : **incomplete information**. The collected feedback instances are then used as the training and development examples for training reward models. For each task prompt x , we only collect fine-grained feedback for *one* model output. Our data collection has IRB approval and is deemed exempt.

Fine-grained human feedback. We instruct workers to identify any error in each model output $y = (a_1, \dots, a_T)$, marking the span of text associated with each identified error type. Formally, we define the set of user-annotated feedback for a task prompt x and model output y as $\mathcal{F} = \{f_i\}$ where each $f_i = \langle c_i, b_i, e_i \rangle$ represents the user-identified span $(a_{b_i}, \dots, a_{e_i})$ of the error category C_{c_i} , where $c_i \in \{1, 2, 3\}$. Importantly,

¹<https://github.com/castorini/pyserini>

we impose three restrictions in the annotation: (1) error spans of category C_1 or C_2 should not overlap with each other; (2) only spans that do not have error C_1 need to be assessed as containing error C_2 or not; (3) C_3 can only apply to whole output sequences. Additionally, we ask workers to mark passage sentences that contain missing information if a C_3 error is annotated. We also ask workers to rewrite y into a corrected version y' that addresses all annotated feedback \mathcal{F} . Details about the feedback collection interface, instructions, and quality control are in Appendix A.2.

To analyze human-human agreement, a subset of 300 examples receive annotations from two distinct workers. We observe that while exact agreement in error span boundaries is low, workers achieve reasonably high agreement on whether a sub-sentence contains C_1 and whether a sentence contains C_2 .² Therefore, we decide to have the density for error type C_1 , C_2 , and C_3 as sub-sentence, sentence and full sequence. We provide more data analysis including human agreement in Appendix A.2.

Preference-based human feedback. For comparison purposes, we follow Ouyang et al. [2022] to separately collect pairwise *human preferences* from the same group of workers. We sample 4 model outputs for each prompt x , which gives 6 pairs of model outputs. We ask the workers to indicate pairwise preferences (ties are allowed) based on all errors they can find in each model output. They are not asked to explicitly annotate these errors.

Annotation costs. On average, both annotation tasks of fine-grained and preference feedback for one question take a worker about 6 minutes to finish. In contrast, Stelmakh et al. [2022] report that they spend about 15 minutes to label a human-written response for each question, which is much more time-consuming than our feedback annotation. On average, we pay \$1.65 per example for both tasks, leading to \$16.50 hourly pay for our workers. We include details of the pay structure in Appendix A.2. We observe that human annotators can reach a higher agreement in each aspect of fine-grained feedback compared to pairwise comparisons because the feedback definitions are more concrete.

6.4.2 Fine-Grained Reward Models

We train three separate reward models R_{ϕ_1} , R_{ϕ_2} , and R_{ϕ_3} for C_1 , C_2 , and C_3 error categories respectively with a density of sub-sentence, sentence, and full sequence, respectively. Since reward models provide

²We use spaCy [Honnibal et al., 2020] to segment generated model outputs into sentences. We then split sentences into sub-sentences using a comma or semicolon.

scalar reward scores and do not perform generation, we use the encoder-only Longformer-base [Beltagy et al., 2020] as our backbone model to handle long input sequences (more details of each reward model are in Appendix A.3).

C_1 : Irrelevance, repetition, or incoherence. R_{ϕ_1} targets to predict whether each sub-sentence in y contains a C_1 type error. We denote $y = (y_1^1, \dots, y_{L_1}^1)$, where y_j^1 is the j th segment at R_{ϕ_1} ’s density (i.e., sub-sentence), with L_1 segments in total. We add a 2-class token-level classification layer (a single feed-forward layer) on the top of the Longformer encoder. The model input has the format of “question: q answer: [sep] y_1^1 [sep] y_2^1 ...”, and we take the classification output at each [sep] token to indicate whether the following y_j^1 contains a C_1 error. We do not add passages in the model input because the detection of C_1 errors does not depend on them. To train R_{ϕ_1} , we apply a token-level classification loss to each [sep] token before y_j^1 , where its gold label g_j is “has error” if there is a $f_i \in \mathcal{F}$ that has $(a_{b_i}, \dots, a_{e_i})$ overlapped with y_j^1 and $c_i = 1$, and “no error” otherwise. When R_{ϕ_1} provides a reward during RL training as in Eq. 6.1, we read a reward $R_{\phi_1}(x, y, j)$ for every y_j^1 given x and y . We define $R_{\phi_1}(x, y, j) = +1$ if R_{ϕ_1} predicts “no error” for y_j^1 and -1 otherwise.

C_2 : Incorrect or unverifiable facts. R_{ϕ_2} is developed for detecting a C_2 error at the sentence level in a similar way. The model input has the format of “question: q context: $p_1 p_2 \dots$ answer: [sep] y_1^2 [sep] y_2^2 ...”, where p ’s denotes the grounding passages and y_j^2 represents the j th sentence. We train R_{ϕ_2} similarly to R_{ϕ_1} , with one exception: as we instruct the workers not to annotate a C_2 error for a span that is already labeled as containing a C_1 error, we do not calculate loss on sentences that are labeled as containing C_1 but not C_2 during R_{ϕ_2} training.

C_3 : Incomplete information. R_{ϕ_3} is trained to measure the information completeness of y , at the full sequence level. Motivated by Li et al. [2019a], R_{ϕ_3} predicts a single scalar reward and is trained with a pairwise comparison loss [Ouyang et al., 2022]:

$$\mathcal{L}_r(\phi) = -\mathbb{E}_{(x, \bar{y}_p, \bar{y}_l) \sim D_p} \left[\log \left(\sigma(R_{\phi_3}(x, \bar{y}_p) - R_{\phi_3}(x, \bar{y}_l)) \right) \right] \quad (6.2)$$

where $R_{\phi_3}(x, y)$ is the scalar output of the reward model for input x and output y ; \bar{y}_p and \bar{y}_l are sampled from the same input x , and \bar{y}_p has less missed information compared with \bar{y}_l ; D_p contains the pairwise comparisons bootstrapped from human feedback on C_3 errors (see details in Appendix A.3).

Preference-based reward model. The preference-based reward model is trained in a similar way to R_{ϕ_3} , with \bar{y}_p representing the human preferred response against \bar{y}_l in the loss function Eq. 6.2. It outputs a scalar score for the given x and y that represents the overall response quality.

6.4.3 Experimental Setup

Compared systems. We compare our proposed method, **FINE-GRAINED RLHF** with the initial T5 policy model trained with 1K examples (**SFT**) and RLHF with holistic preference-based rewards (**Preference RLHF**). The reward models used in RLHF experiments are trained on 2.8K examples with annotated feedback (but no gold human response). For analysis, we also use the human gold responses of all training examples to finetune a fully supervised T5 model (**SFT-Full**). Notice that SFT-Full requires much higher annotation cost because it takes longer (15 minutes per example [Stelmakh et al., 2022]) for annotators to draft long-form responses.

Implementation details. Our policy model is based on T5-large [Raffel et al., 2020] and is supervised finetuned on 1K training examples, as explained in §6.4. During RL exploration, we use top-k ($k = 20$) sampling decoding with temperature = 0.7, which is set based on previous RLHF work [Ramamurthy et al., 2023]. The value model used during RL training is initialized with T5-base due to GPU memory constraint. The reward model weights we used in FINE-GRAINED RLHF are $w_1 = 0.3, w_2 = 0.5, w_3 = 0.3$, unless otherwise specified. Although we use three reward models during RL training, we only observe very small relative additional cost (roughly 1% training time) compared to preference RLHF. During inference, we use greedy decoding to generate responses. We report more details including RL training parameters in Appendix A.1. All scores reported are averaged over 3 independent runs.

Evaluation. We conduct both human and automatic evaluation. Human evaluation is run on 200 randomly sampled test set examples of QA-FEEDBACK to compare *Fine-Grained RLHF* with all baselines. Each model output is sampled from inference results of 3 training runs. We use the same protocol of feedback collection to have the same set of workers annotate spans in each model output that contain (1) irrelevance, repetition, or incoherence error (**rel.**) and (2) incorrect or unverifiable facts (**fact.**). They are also asked to compare the information completeness (**comp.**) for each output pair. To report evaluation scores for *rel.* and *fact.* error spans, we first map them to their corresponding error type density (sub-sentence and sentence).

Then we report the error rate for each error type, measured as the percentage of sub-sentences that contains this type of error. Since spans with *rel.* error are not checked for *fact.* error (discussed in §6.4.1), we exclude sub-sentences with only *rel.* error when report the error rate of *fact.* error. For automatic evaluation, we report RougeLSum [Lin, 2004] as used for the original ASQA data, as well as the score from each fine-grained reward model (R_{ϕ_1} , R_{ϕ_2} , and R_{ϕ_3}). Specifically, we report the percentage of all sub-sentences (or sentences) in the test set predicted as “no error” by R_{ϕ_1} (or R_{ϕ_2}). For R_{ϕ_3} , we report the averaged output score for all test examples.

6.4.4 Main Results

Figure 6.3 shows the human evaluation results for *rel.* and *fact.* error types. Table 6.2 shows the human pairwise comparison results for information completeness (*comp.*).

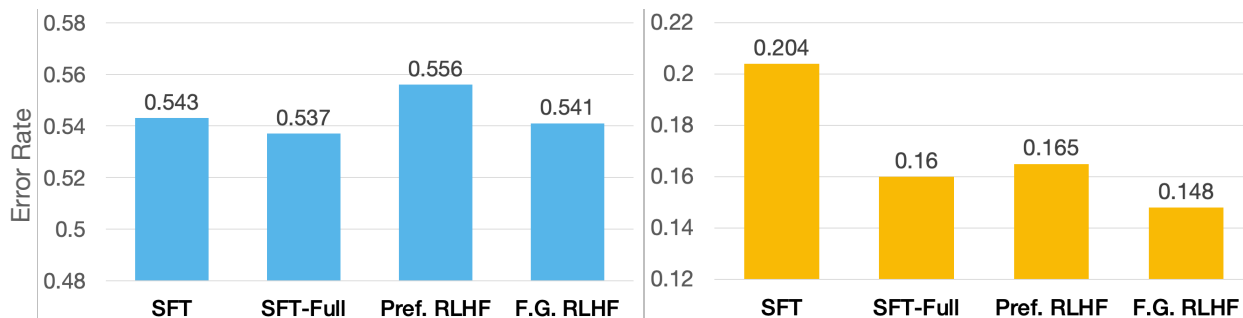


Figure 6.3: Human evaluation on *rel.* (left) and *fact.* (right) error, measured by % of sub-sentences that contain the error type (\downarrow).

Ours vs.	Win	Tie	Lose
SFT	23.0%	65.5%	11.5%
SFT-Full	22.0%	61.0%	17.0%
Pref. RLHF	19.5%	71.0%	9.5%

Table 6.2: Human pairwise comparison on *information completeness (comp.)*, where win/lose refers to FINE-GRAINED RLHF.

FINE-GRAINED RLHF outperforms SFT and Preference RLHF on all error types. Figure 6.3 and Table 6.2 show that our FINE-GRAINED RLHF leads to generation that is much more factually correct and contains more complete information, compared to all other systems. It generates fewer irrelevance, repetition, and incoherence errors, compared with SFT and Preference RLHF. In the meantime, Preference RLHF,

despite greatly reducing factual errors compared to the initial policy model SFT, generates even more irrelevance, repetition, and incoherence errors than SFT. FINE-GRAINED RLHF outperforms Preference RLHF potentially due to more specific and localized training signals. In addition, we ask annotators to compare the overall generation quality of FINE-GRAINED RLHF and preference RLHF. Although Preference RLHF is trained directly with such preference feedback, FINE-GRAINED RLHF was rated better than Preference RLHF in 30.5% of all examples and worse in 24.5% of examples, although it is not statistically significant according to a binomial model based on win and loss counts. The annotators indicate a tie in the remaining 45% of cases. Surprisingly, FINE-GRAINED RLHF outperforms SFT-Full with more factual and complete generation, despite a much lower annotation cost.

RLHF is particularly effective in reducing factual errors. Figure 6.3 shows that both FINE-GRAINED RLHF and Preference RLHF are effective in reducing factual errors in model generation. Meanwhile, we see little or no improvement in reducing irrelevance, repetition, or incoherence errors. We provide more in-depth analysis for this observation in §6.4.5.

Table 6.3 shows automatic scores on the QA-FEEDBACK test set, which show similar trends as human evaluation in terms of system comparisons, while all four systems achieve similar Rouge scores.

	rel. $R_{\phi_1}(\uparrow)$	fact. $R_{\phi_2}(\uparrow)$	comp. $R_{\phi_3}(\uparrow)$	Rouge(\uparrow)
SFT-Full	0.508	0.756	0.044	49.63
SFT	0.513	0.749	-0.053	48.96
+ Pref. RLHF	0.482	0.781	0.101	49.84
+ E.G. RLHF	0.513	0.816	0.139	49.93

Table 6.3: Automatic evaluation on the QA-FEEDBACK test set.

	rel. $R_{\phi_1}(\uparrow)$	fact. $R_{\phi_2}(\uparrow)$	comp. $R_{\phi_3}(\uparrow)$	Rouge(\uparrow)	avg. len
Short	0.637	0.760	-0.231	48.99	74.92
Medium	0.513	0.816	0.139	49.93	98.66
Long	0.425	0.860	0.241	48.72	109.63

Table 6.4: Automatic evaluation results (test set) of FINE-GRAINED RLHF trained with different reward model weight configurations.

6.4.5 LM Customization with FINE-GRAINED RLHF

Since we use multiple reward models in FINE-GRAINED RLHF, adjusting their weights (see Eq. 6.1) during RL may lead to different LM behaviors. For example, adding more weight to a reward model associated with one specific desired behavior type (e.g., information completeness) may lead the generation more towards that behavior type compared to others (e.g., information relevance). This flexibility can potentially fit users with diverse needs. Therefore, in this section, we explore FINE-GRAINED RLHF’s ability to customize the

LM behavior.

LM customization. As in Table 6.4, we explore three configurations of reward model weights (w_1 , w_2 , and w_3 for R_{ϕ_1} , R_{ϕ_2} , and R_{ϕ_3}) and name them ‘short’, ‘medium’, and ‘long’ according to the LM’s average generation length. For simplicity, we fix $w_2 = 0.5$ and $w_3 = 0.3$, and use 0.4, 0.3, and 0.2 for w_1 , which leads to ‘short’, ‘medium’, and ‘long’ generation outputs respectively. We manually inspect 30 random examples and observe that (1) ‘short’ generates more relevant content, but is less factual and complete; (2) ‘long’, in contrast, gives the most factual and complete generation. This reflects that the LM is referencing a large amount of content from passages; (3) The ‘medium’ configuration balances the three rewards and has the highest Rouge score. 24/30 examples follow the above rule. Qualitative analysis and examples of LM customization are in Appendix A.4.

Trade-off between error types. We observe that a higher w_1 leads to a bigger *rel.* reward, smaller *fact.* and *comp.* rewards, and shorter generated outputs. One interpretation is that R_{ϕ_1} penalizes text spans that are irrelevant to the questions. As such, it encourages answering the question directly and penalizes referencing passages and generating auxiliary information. This reduces the model generation length and information completeness, and introduces more factual errors.

6.4.6 Analysis

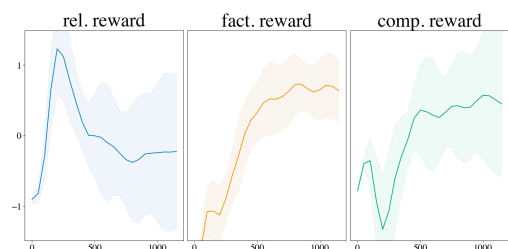


Figure 6.4: Dynamics of each type of reward during training (reward vs. training steps). All rewards are z-normalized.

	rel. $R_{\phi_1}(\uparrow)$	fact. $R_{\phi_2}(\uparrow)$	comp. $R_{\phi_3}(\uparrow)$	Rouge(\uparrow)	avg. len
SFT	0.514	0.735	0.065	43.13	96.69
F.G. RLHF	0.516	0.825	0.266	44.29	101.76
w/o. R_{ϕ_1}	0.249	0.771	0.742	38.52	179.31
w/o. R_{ϕ_2}	0.716	0.640	-0.177	43.18	78.08
w/o. R_{ϕ_3}	0.565	0.799	0.123	43.61	93.92

Table 6.5: Ablation of reward models on the development set. R_{ϕ_1} , R_{ϕ_2} , and R_{ϕ_3} correspond to the reward model for relevance, factuality, and information completeness.

Reward models are competing against each other. In the prior section, we find that there is a trade-off between error types. To further look into this phenomenon, we explore the dynamics of each reward model during training. Figure 6.4 shows each reward model’s rewards on the development set during training. All rewards are z-normalized for visualization. We see that the *fact.* reward is consistently increasing. The *rel.*

reward increases rapidly in the first 250 steps and then starts decreasing, while the *comp.* reward exhibits an opposite trend, decreasing at first and then starting to increase. As discussed earlier, one interpretation is that relevance (precision) and information completeness (recall) can be adversarial objectives, so the rewards are competing. The three rewards reach an equilibrium point in later steps.

Ablation: Does the LM learn from all reward models? What if we remove one reward model? Table 6.5 explores the policy LM behavior when one of the three reward models is removed during training. Qualitative examples are in Appendix A.4. First, we observe that the corresponding reward decreases dramatically when the model is removed. When the *rel.* reward model (R_{ϕ_1}) is removed, the outputs become extremely long and the *comp.* reward is extremely high. We observe the outputs and find the model is copying a lot of content from the passages. When the *fact.* reward model (R_{ϕ_2}) is removed, the *rel.* reward becomes the highest. We observe that the LM tends to answer the question directly and not reference the passages, which causes a lot of hallucinations. When the *comp.* reward model (R_{ϕ_3}) is removed, the outputs are concise and factual but not providing all relevant information to the question. Thus, it has lower information completeness and Rouge score compared with the LM trained with all reward models.

Reward model performance. We report and analyze the performance of each reward model in predicting its corresponding error category. The *rel.* reward model R_{ϕ_1} has a binary classification accuracy of 69.6, and an F1 score (for the “has error” class) of 68.5 on model-generated sub-sentences from the development set. We sample 20 sub-sentences where R_{ϕ_1} predicts the opposite of the human label, and observe that all of them either 1) contain relevant auxiliary information and are marked as “no error” by humans, or 2) are marked as irrelevant by humans but provide closely related background information to the question. In other words, R_{ϕ_1} is mostly struggling with predicting the relevance of auxiliary information, and it rarely fails to predict a direct answer as “no error”.

The *fact.* reward model R_{ϕ_2} has an accuracy of 77.8 and an F1 score of 67.5. We sample 20 sentences where R_{ϕ_2} makes a prediction mistake and we observe that the mistakes often happen when the generated sentence is highly abstractive instead of directly copying information from the passage. We also observe that more than 80% of human labeled factual errors occur when the model generates a direct answer (not auxiliary information) that contains hallucinated information or a random entity from a passage. We notice that R_{ϕ_2} correctly captures more than 80% of such errors.

The *comp.* reward model R_{ϕ_3} has an accuracy of 70.9 in pairwise comparison. In contrast, the preference-based reward model only reaches an accuracy of 57.2. This helps confirm our intuition that assessing long-form generation outputs holistically can be more ambiguous and subjective than evaluating the outputs with a focus on a specific undesired behavior type.

Comparison with ChatGPT responses. We experiment with answering the questions with ChatGPT.³ To familiarize ChatGPT with the style of our LFQA task, we prompt it with the task instruction and a single random QA example (due to length limitation). ChatGPT achieves a RougeLSum score of 40.92 on the test set, which is much lower than our models. We do not use our trained reward models to evaluate ChatGPT outputs because reward models trained on T5-large may not generalize well to ChatGPT. We instead manually inspect the ChatGPT responses, and observe that they are mostly concise and factual, yet lack the auxiliary information necessary to clarify ambiguous questions. Qualitative examples are in Appendix A.4. This shows the difficulty for ChatGPT in learning user-desired behaviors through simple prompting.

6.5 Discussion

6.5.1 Broader Impacts

We propose the FINE-GRAINED RLHF framework that can incorporate multiple reward models to provide dense rewards for RL training with costs similar to preference-based feedback, which leads to LM outputs that are optimized towards such rewards. Our framework can be applied to any text generation task, thereby enhancing LM performance by offering more nuanced guidance than holistic feedback. The key advantages of the FINE-GRAINED RLHF framework are two-fold:

Flexibility. Our framework significantly expands the versatility of reward models for RLHF. For example, future work involving fact-checking, sentiment classification, toxicity detection, among others, can all be incorporated within this framework. LMs can be trained against all these reward models via FINE-GRAINED RLHF.

Controllability. Having multiple reward models that stand for different feedback types allows the end user to exert greater control over RL training (e.g., through different combinations of reward model weights; see

³We used gpt-3.5-turbo.

details in § 6.4.5). This leads to customized model behaviors, a benefit particularly valuable for applications like educational tools where model personalization is crucial.

6.5.2 Limitations

One major limitation of our framework comes from the additional compute cost of getting *fine-grained* rewards, compared to RLHF with a holistic reward. For instance, in the detoxification task, we need to make multiple PERSPECTIVE API calls for each model output depending on how many sentences are generated, while RLHF with a holistic reward only requires one. In the long-form QA task, we need to calculate a dense reward from multiple reward models, which takes more compute time and GPU memory than a single reward model. However, since running reward models accounts for only a small percentage of the total training time, training FINE-GRAINED RLHF requires less than an additional 5% of the time compared to preference-based RLHF.

Another limitation is that different tasks may have different definitions of fine-grained feedback in terms of the feedback types and the density level of each type. Therefore, defining feedback that is well-suited for a task, easy to annotate, and training reward models accordingly requires non-trivial manual effort.

6.6 Summary

In this work, we introduce FINE-GRAINED RLHF, a framework that enables LMs to learn from multiple fine-grained reward models trained from human feedback, where each reward model detects a specific error category and provides dense rewards. We conduct experimental analysis on two text generation tasks to illustrate the performance gain of FINE-GRAINED RLHF than RLHF over holistic rewards, supported by both automatic and human evaluation. Furthermore, we show that an LM can be customized for specific needs using different combinations of fine-grained reward models.

Chapter 7

Conclusion

7.1 Summary and Contributions

In this thesis, we propose solutions to address key challenges towards building an interactive agent to support human users’ information-seeking needs. The three challenges and our corresponding contributions are summarized as follows.

7.1.1 Knowledge Retrieval and Identification in Conversations

We introduce DIALKI, a *knowledge identification* model to address the problem of locating relevant information from long documents in a conversational context (Chapter 3). Specifically, we introduce a multi-task learning framework that models the dialogue-document interactions via an auxiliary task of history knowledge prediction and a knowledge contextualization mechanism. DIALKI advances the state of the art in knowledge identification tasks for two conversational datasets, with more than 60% and 20% gains over previous work. DIALKI also won the first place in the knowledge identification subtask in the DialDoc Shared Task [Feng et al., 2022].

Further, we introduce CONQRR to address the problem of retrieving relevant passages from a large corpus. CONQRR is the first RL-based QR model that can be adapted to and optimized towards any off-the-shelf retriever for conversational QA (Chapter 4). We demonstrate that CONQRR achieves state-of-the-art results with off-the-shelf retrievers on QReCC with conversations from three sources and is effective for two

retrievers, including BM25 and a dual encoder model. Our analysis shows CONQRR trained with *no human rewrite supervision* provides better retrieval results than strong baselines trained with full supervision, and it is robust to out-of-domain dialogues, topic shifts and long dialogue contexts. We conduct a novel quantitative study to analyze the limitations and utility of human rewrites in retrieval performance, which are largely unexplored in prior work. We observe that human rewrites sometimes omit information from the dialogue context that is useful to the retriever, which motivates the development of CONQRR, but they can still serve as useful supervision for conversations with frequent topic shifts.

7.1.2 Interactive and Collaborative Information Seeking

As most existing studies either fail to or artificially incorporate agent-side initiatives to collaborate with the human user in a information-seeking conversation, we collect INSCIT, a dataset for **Information-Seeking Conversations with mixed-initiative Interactions** (Chapter 5). It contains 4.7K user-agent turns from 805 human-human conversations where the agent searches over Wikipedia and either directly answers, asks for clarification, or provides relevant information to address user queries. The data supports two subtasks, evidence passage identification and response generation, as well as a human evaluation protocol to assess model performance. We report results of two systems based on state-of-the-art models of conversational knowledge identification and open-domain question answering. Since both systems significantly underperform humans, we conduct systematic analysis to suggest ways for improvement in future studies.

7.1.3 Response Generation Informed by Human-centered Objectives

We introduce FINE-GRAINED RLHF, a framework that enables training and learning from reward functions that are fine-grained in two respects: (1) density, providing a reward after every segment (e.g., a sentence) is generated; and (2) incorporating multiple reward models associated with different feedback types (Chapter 6). We collect QA-FEEDBACK, a dataset of both coarse-grained and fine-grained human feedback for long-form question answering tasks. We conduct experiments on detoxification and long-form question answering to illustrate how learning with such reward functions leads to improved performance, supported by both automatic and human evaluation. Additionally, we show that LM behaviors can be customized using different combinations of fine-grained reward models.

The contributions in this thesis include modeling frameworks, datasets, evaluation frameworks, and learning strategies. We have released all code and data for all our work discussed in this thesis, except for CONQRR, which was done in the industry lab. We believe the challenges discussed in this thesis are crucial to enhance the experience of an information-seeking user when interacting with the agent model and to optimize the helpfulness of information being returned to the user. We hope that our proposed approaches can be useful for future research with similar goals.

7.2 Future Directions

We discuss a few important future directions to further improve agent models for human-centered interactive information-seeking.

Interactive and collaborative agents. We show in Chapter 5 that there is still improvement room for the model to learn when and how to use non-direct response strategies. Given the small size of our training data, a future direction is to explore transfer learning using existing information-seeking conversation or question answering resources. INSCIT focuses on different strategies that can be adopted by the agent to better address user requests in a conversational question answering setting, assuming the user will either ask an information-seeking question or provide a clarification to the agent. Exploring more user-side strategies would be interesting for handling system errors and other types of conversations (e.g. negotiations).

A broader scope of information sources. CONQRR explores how to use passage retrievers like a search engine in an information-seeking conversation effectively. Exploring how to broaden the scope of information sources can be a very interesting direction towards building practical information-seeking applications. People do not always go to a search engine when they look for information. For example, sometimes they go directly to certain websites for specific types of information. Or they can search over local files stored on the server or database systems. It could be an interesting future exploration to train the agent model to interact with a variety of knowledge sources or tools (e.g., functional APIs) to fulfill users' information need. For example, future studies can investigate how to teach a model to differentiate the query types and look for useful information from different sources accordingly.

Learning from natural human feedback. In FINE-GRAINED RLHF, we carefully control the quality of annotated feedback, which is then used to train reward models for RL. In practice, when a deployed model is released to the public, end users do not always give clean feedback. Therefore, how to obtain effective learning signals from noisy user feedback in the real world still needs further investigation.

Exploring how to elicit, identify and learn from natural language human feedback is another promising direction. Humans give and receive feedback to and from others all the time through communication, sometimes even when they are not specifically asking for it. This also adds to the benefit of building conversational agent models as conversing with users is essential for getting such feedback. Natural language feedback can be much easier to collect compared to specifically asking users to provide structured feedback for the model’s continual learning. Designing mechanisms to improve user engagement or explaining the agent’s decision making process via conversation can further trigger more useful feedback. In the meantime, exploring how to identify indirect feedback and convert it into effective learning signals is a near-future opportunity.

Personalized agents. Training a model to adapt to users’ own feedback/objectives and habits of tool uses can lead to personalized agents, which is a future direction. Ideally, a personal agent or assistant should go beyond the scope of information seeking. In many scenarios, the agent should also be able to provide both emotional support and to perform a variety of tasks. To perform a given task, the agent should be able to do planning and take actions by leveraging tools. Based on reward from the working environment, which can be either positive or negative, the agent can make refinements to their planning. If the agent gets stuck, especially for tasks that are never seen before or depend on unknown personal preference, they can ask the human user for feedback or guidance, or collaborate with them. They would also store their interaction experience with both the environment and the user, the user’s preference and habits into their memory for future references. Ideally, the agent would have a learning curve at the beginning to adapt to the user preference or habits, and then the agent would gradually provide more satisfying interaction experience for the user. It could also be important for an agent to learn to balance when to follow the user and when to hold their own opinion, especially on topics that relate to safety and ethical issues.

Responsible information-seeking. Information sources used in practice (e.g., the whole web) can often contain less trustworthy information. In such cases, retrieving evidence passages containing different answers and predicting the trustworthiness of each answer based on all such retrieved passages can be a promising direction to prevent misinformation.

Another direction that is highly relevant to responsible information-seeking lies in the design of evaluation metrics. Well-defined evaluation metrics allow developers to measure the performance of their systems accurately and identify areas for improvement. Reliable and transparent evaluation metrics build user trust in information-seeking systems. This thesis follows previous studies to evaluate the model performance when given a fixed human-human dialogue context. As pointed out by Li et al. [2022], an interactive dialogue system often needs to handle dialogue contexts containing errors made by the model itself. Therefore, it is important for future work to develop new methods for automatic evaluation and scalable human evaluation in the interactive setting. Fine-grained evaluation on bias, fairness and hallucination of information-seeking systems is also crucial towards responsible information-seeking.

Bibliography

Perspective API, <https://github.com/conversationai/perspectiveapi>.

Vaibhav Adlakha, Shehzaad Dhuliawala, Kaheer Suleman, Harm de Vries, and Siva Reddy. 2022. Topic-OCQA: Open-domain conversational question answering with topic switching. In *Transactions of the Association for Computational Linguistics*, volume 10, pages 468–483.

Mohammad Aliannejadi, Julia Kiseleva, Aleksandr Chuklin, Jeff Dalton, and Mikhail Burtsev. 2021. Building and evaluating open-domain dialogue corpora with clarifying questions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4473–4484, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Raviteja Anantha, Svitlana Vakulenko, Zhucheng Tu, Shayne Longpre, Stephen Pulman, and Srinivas Chappidi. 2021. Open-domain question answering goes conversational via question rewriting. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 520–534, Online. Association for Computational Linguistics.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022a. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

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

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder,

- Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. MS MARCO: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.
- Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.
- Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.
- Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep communicating agents for abstractive summarization. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1662–1675, New Orleans, Louisiana. Association for Computational Linguistics.
- Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2020. Evaluation of text generation: A survey. *arXiv preprint arXiv:2006.14799*.
- Angelica Chen, Jérémy Scheurer, Tomasz Korbak, Jon Ander Campos, Jun Shern Chan, Samuel R Bowman, Kyunghyun Cho, and Ethan Perez. 2023. Improving code generation by training with natural language feedback. *arXiv preprint arXiv:2303.16749*.
- Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.
- Hao Cheng, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2020. Probabilistic assumptions matter: Improved models for distantly-supervised document-level question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5657–5667, Online. Association for Computational Linguistics.

- Hao Cheng, Xiaodong Liu, Lis Pereira, Yaoliang Yu, and Jianfeng Gao. 2021. Posterior differential regularization with f-divergence for improving model robustness. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1078–1089, Online. Association for Computational Linguistics.
- 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 *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184, Brussels, Belgium. Association for Computational Linguistics.
- Eunsol Choi, Jennimaria Palomaki, Matthew Lamm, Tom Kwiatkowski, Dipanjan Das, and Michael Collins. 2021. Decontextualization: Making Sentences Stand-Alone. *Transactions of the Association for Computational Linguistics*, 9:447–461.
- Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.
- Jeffrey Dalton, Chenyan Xiong, Vaibhav Kumar, and Jamie Callan. 2020. Cast-19: A dataset for conversational information seeking. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '20*, page 1985–1988, New York, NY, USA. Association for Computing Machinery.
- Bhavana Dalvi Mishra, Oyvind Tafjord, and Peter Clark. 2022. Towards teachable reasoning systems: Using a dynamic memory of user feedback for continual system improvement. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9465–9480, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*.

- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- 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*.
- Yao Dou, Maxwell Forbes, Rik Koncel-Kedziorski, Noah A. Smith, and Yejin Choi. 2022. Is GPT-3 text indistinguishable from human text? Scarecrow: A framework for scrutinizing machine text. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7250–7274, Dublin, Ireland. Association for Computational Linguistics.
- Ahmed Elgohary, Denis Peskov, and Jordan Boyd-Graber. 2019. Can you unpack that? learning to rewrite questions-in-context. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5918–5924, Hong Kong, China. Association for Computational Linguistics.
- Song Feng, Siva Patel, and Hui Wan. 2022. DialDoc 2022 shared task: Open-book document-grounded dialogue modeling. In *Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering*, pages 155–160, Dublin, Ireland. Association for Computational Linguistics.
- Song Feng, Hui Wan, Chulaka Gunasekara, Siva Patel, Sachindra Joshi, and Luis Lastras. 2020. doc2dial: A goal-oriented document-grounded dialogue dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8118–8128, Online. Association for Computational Linguistics.
- Adam Fisch, Kenton Lee, Ming-Wei Chang, Jonathan Clark, and Regina Barzilay. 2020. CapWAP: Image

- captioning with a purpose. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8755–8768, Online. Association for Computational Linguistics.
- Yao Fu, Litu Ou, Mingyu Chen, Yuhao Wan, Hao Peng, and Tushar Khot. 2023. Chain-of-thought hub: A continuous effort to measure large language models’ reasoning performance. *arXiv preprint arXiv:2305.17306*.
- Deep Ganguli, Amanda Askell, Nicholas Schiefer, Thomas Liao, Kamilè Lukošiūtė, Anna Chen, Anna Goldie, Azalia Mirhoseini, Catherine Olsson, Danny Hernandez, et al. 2023. The capacity for moral self-correction in large language models. *arXiv preprint arXiv:2302.07459*.
- Chang Gao, Wenxuan Zhang, and Wai Lam. 2022. UniGDD: A unified generative framework for goal-oriented document-grounded dialogue. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 599–605, Dublin, Ireland. Association for Computational Linguistics.
- Jianfeng Gao, Michel Galley, and Lihong Li. 2018. Neural approaches to conversational AI. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 2–7, Melbourne, Australia. Association for Computational Linguistics.
- Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6465–6488, Singapore. Association for Computational Linguistics.
- Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369, Online. Association for Computational Linguistics.
- Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen tau Yih, and Michel Galley. 2018. A knowledge-grounded neural conversation model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pages 5110–5117.

- Amelia Glaese, Nat McAleese, Maja Trębacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, et al. 2022. Improving alignment of dialogue agents via targeted human judgements. *arXiv preprint arXiv:2209.14375*.
- Meiqi Guo, Mingda Zhang, Siva Reddy, and Malihe Alikhani. 2021. Abg-coQA: Clarifying ambiguity in conversational question answering. In *3rd Conference on Automated Knowledge Base Construction*.
- Alexandru Balan Gustavo Penha and Claudia Hauff. 2019. Introducing MANtIS: a novel multi-domain information seeking dialogues dataset. *arXiv preprint arXiv:1912.04639*.
- Braden Hancock, Antoine Bordes, Pierre-Emmanuel Mazare, and Jason Weston. 2019. Learning from dialogue after deployment: Feed yourself, chatbot! In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3667–3684, Florence, Italy. Association for Computational Linguistics.
- Tatsunori B. Hashimoto, Hugh Zhang, and Percy Liang. 2019. Unifying human and statistical evaluation for natural language generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1689–1701, Minneapolis, Minnesota. Association for Computational Linguistics.
- Mikael Henaff, Jason Weston, Arthur Szlam, Antoine Bordes, and Yann LeCun. 2017. Tracking the world state with recurrent entity networks. In *International Conference on Learning Representations*.
- Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.
- Mohit Iyyer, Wen-tau Yih, and Ming-Wei Chang. 2017. Search-based neural structured learning for sequential question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1821–1831, Vancouver, Canada. Association for Computational Linguistics.
- Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the*

- Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.
- Dietmar Jannach, Ahtsham Manzoor, Wanling Cai, and Li Chen. 2021. A survey on conversational recommender systems. *ACM Comput. Surv.*, 54(5).
- Dongyeop Kang, Anusha Balakrishnan, Pararth Shah, Paul Crook, Y-Lan Boureau, and Jason Weston. 2019. Recommendation as a communication game: Self-supervised bot-play for goal-oriented dialogue. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1951–1961, Hong Kong, China. Association for Computational Linguistics.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.
- Byeongchang Kim, Jaewoo Ahn, and Gunhee Kim. 2020. Sequential latent knowledge selection for knowledge-grounded dialogue. In *International Conference on Learning Representations*.
- Sungdong Kim and Gangwoo Kim. 2022. Saving dense retriever from shortcut dependency in conversational search. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10278–10287, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Natalia Konstantinova and Constantin Orasan. 2013. Interactive question answering. *Emerging Applications of Natural Language Processing: Concepts and New Research*, pages 149–169.
- Tomasz Korbak, Kejian Shi, Angelica Chen, Rasika Bhalerao, Christopher L Buckley, Jason Phang, Samuel R Bowman, and Ethan Perez. 2023. Pretraining language models with human preferences. In *Proceedings of the 40th International Conference on Machine Learning*, pages 17506–17533.
- Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. 2021. GeDi: Generative discriminator guided sequence generation. In

- Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4929–4952, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Kalpesh Krishna, Erin Bransom, Bailey Kuehl, Mohit Iyyer, Pradeep Dasigi, Arman Cohan, and Kyle Lo. 2023. LongEval: Guidelines for human evaluation of faithfulness in long-form summarization. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1650–1669, Dubrovnik, Croatia. Association for Computational Linguistics.
- Vaibhav Kumar and Jamie Callan. 2020. Making information seeking easier: An improved pipeline for conversational search. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3971–3980, Online. Association for Computational Linguistics.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, and Chris Alberti et al. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Huihan Li, Tianyu Gao, Manan Goenka, and Danqi Chen. 2022. Ditch the gold standard: Re-evaluating conversational question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8074–8085, Dublin, Ireland. Association for Computational Linguistics.
- Margaret Li, Jason Weston, and Stephen Roller. 2019a. Acute-eval: Improved dialogue evaluation with optimized questions and multi-turn comparisons. *arXiv preprint arXiv:1909.03087*.
- Raymond Li, Samira Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards deep conversational recommendations. In *Proceedings of the 32nd International Conference*

- on *Neural Information Processing Systems*, NIPS'18, page 9748–9758, Red Hook, NY, USA. Curran Associates Inc.
- Zekang Li, Cheng Niu, Fandong Meng, Yang Feng, Qian Li, and Jie Zhou. 2019b. Incremental transformer with deliberation decoder for document grounded conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 12–21, Florence, Italy. Association for Computational Linguistics.
- Rongzhong Lian, Min Xie, Fan Wang, Jinhua Peng, and Hua Wu. 2019. Learning to select knowledge for response generation in dialog systems. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5081–5087. International Joint Conferences on Artificial Intelligence Organization.
- Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2021. Contextualized query embeddings for conversational search. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1004–1015, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Sheng-Chieh Lin, Jheng-Hong Yang, Rodrigo Nogueira, Ming-Feng Tsai, Chuan-Ju Wang, and Jimmy Lin. 2020a. Conversational question reformulation via sequence-to-sequence architectures and pretrained language models. *arXiv preprint arXiv:2004.01909*.
- Xiexiong Lin, Weiyu Jian, Jianshan He, Taifeng Wang, and Wei Chu. 2020b. Generating informative conversational response using recurrent knowledge-interaction and knowledge-copy. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 41–52, Online. Association for Computational Linguistics.
- Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 2021. DExperts: Decoding-time controlled text generation with experts and anti-experts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the*

- 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6691–6706, Online. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Zeming Liu, Haifeng Wang, Zheng-Yu Niu, Hua Wu, Wanxiang Che, and Ting Liu. 2020. Towards conversational recommendation over multi-type dialogs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1036–1049, Online. Association for Computational Linguistics.
- Ximing Lu, Sean Welleck, Jack Hessel, Liwei Jiang, Lianhui Qin, Peter West, Prithviraj Ammanabrolu, and Yejin Choi. 2022. Quark: Controllable text generation with reinforced unlearning. In *Advances in Neural Information Processing Systems*, volume 35, pages 27591–27609. Curran Associates, Inc.
- Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics*, 9:329–345.
- Aman Madaan, Niket Tandon, Peter Clark, and Yiming Yang. 2022. Memory-assisted prompt editing to improve GPT-3 after deployment. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2833–2861, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020. Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4984–4997, Online. Association for Computational Linguistics.
- Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, et al. 2022. Teaching language models to support answers with verified quotes. *arXiv preprint arXiv:2203.11147*.

- Sewon Min, Danqi Chen, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019. Knowledge guided text retrieval and reading for open domain question answering. *arXiv preprint arXiv:1911.03868*.
- Sewon Min, Kalpesh Krishna, Xinxu Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.
- Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797, Online. Association for Computational Linguistics.
- Nikita Moghe, Siddhartha Arora, Suman Banerjee, and Mitesh M. Khapra. 2018. Towards exploiting background knowledge for building conversation systems. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2322–2332, Brussels, Belgium. Association for Computational Linguistics.
- Reiichiro Nakano, Jacob Hilton, S. Arun Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2021. Webgpt: Browser-assisted question-answering with human feedback. *ArXiv*, abs/2112.09332.
- Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, and Yinfei Yang. 2022. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022.

- Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Debjit Paul, Mete Ismayilzada, Maxime Peyrard, Beatriz Borges, Antoine Bosselut, Robert West, and Boi Faltings. 2023. Refiner: Reasoning feedback on intermediate representations. *ArXiv*, abs/2304.01904.
- Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*.
- John Pavlopoulos, Jeffrey Sorensen, Lucas Dixon, Nithum Thain, and Ion Androutsopoulos. 2020. Toxicity detection: Does context really matter? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4296–4305, Online. Association for Computational Linguistics.
- Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.
- Lianhui Qin, Michel Galley, Chris Brockett, Xiaodong Liu, Xiang Gao, Bill Dolan, Yejin Choi, and Jianfeng Gao. 2019. Conversing by reading: Contentful neural conversation with on-demand machine reading. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5427–5436, Florence, Italy. Association for Computational Linguistics.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*. Accessed 22 March 2021.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*

- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.
- Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. 2023. Is reinforcement learning (not) for natural language processing: Benchmarks, baselines, and building blocks for natural language policy optimization. In *International Conference on Learning Representations (ICLR)*.
- Marc’Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.
- 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.
- Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1179–1195.
- V. Rieser and O. Lemon. 2009. Does this list contain what you were searching for? Learning adaptive dialogue strategies for interactive question answering. *Natural Language Engineering*, 15(1):55–72.
- Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.
- Marzieh Saeidi, Max Bartolo, Patrick Lewis, Sameer Singh, Tim Rocktäschel, Mike Sheldon, Guillaume Bouchard, and Sebastian Riedel. 2018. Interpretation of natural language rules in conversational machine reading. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2087–2097, Brussels, Belgium. Association for Computational Linguistics.
- Amrita Saha, Vardaan Pahuja, Mitesh M. Khapra, Karthik Sankaranarayanan, and Sarath Chandar. 2018. Complex sequential question answering: towards learning to converse over linked question answer pairs

- with a knowledge graph. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*.
- Jérémy Scheurer, Jon Ander Campos, Jun Shern Chan, Angelica Chen, Kyunghyun Cho, and Ethan Perez. 2022. Training language models with natural language feedback. *arXiv preprint arXiv:2204.14146*.
- Jérémy Scheurer, Jon Ander Campos, Tomasz Korbak, Jun Shern Chan, Angelica Chen, Kyunghyun Cho, and Ethan Perez. 2023. Training language models with language feedback at scale. *arXiv preprint arXiv:2303.16755*.
- John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2016. High-dimensional continuous control using generalized advantage estimation. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.
- Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4596–4604. PMLR.
- Weiyan Shi, Emily Dinan, Kurt Shuster, Jason Weston, and Jing Xu. 2022. When life gives you lemons, make cherryade: Converting feedback from bad responses into good labels. *arXiv preprint arXiv:2210.15893*.
- Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803, Punta Cana, Dominican Republic. Association for Computational Linguistics.

- Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. ASQA: Factoid questions meet long-form answers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8273–8288, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems*, volume 33, pages 3008–3021. Curran Associates, Inc.
- C. A. Thompson, M. H. Goker, and P. Langley. 2004. A personalized system for conversational recommendations. *Journal of Artificial Intelligence Research*, 21:393–428.
- Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.
- Svitlana Vakulenko, Shayne Longpre, Zhucheng Tu, and Raviteja Anantha. 2021. Question rewriting for conversational question answering. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining, WSDM '21*, page 355–363, New York, NY, USA. Association for Computing Machinery.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz

- Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Sergey Volokhin, Joyce Ho, Oleg Rokhlenko, and Eugene Agichtein. 2021. You sound like someone who watches drama movies: Towards predicting movie preferences from conversational interactions. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3091–3096, Online. Association for Computational Linguistics.
- Ellen M. Voorhees and Dawn M. Tice. 2000. The TREC-8 question answering track. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, Athens, Greece. European Language Resources Association (ELRA).
- Nikos Voskarides, Dan Li, Pengjie Ren, Evangelos Kanoulas, and Maarten de Rijke. 2020. Query resolution for conversational search with limited supervision. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '20*, page 921–930, New York, NY, USA. Association for Computing Machinery.
- N. Webb and B. Webber. 2009. Special issue on interactive question answering: Introduction. *Natural Language Engineering*, 15(1):1–8.
- Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 438–449, Valencia, Spain. Association for Computational Linguistics.
- Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. 2013. The dialog state tracking challenge. In *Proceedings of the SIGDIAL 2013 Conference*, pages 404–413, Metz, France. Association for Computational Linguistics.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen,

- Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Jeff Wu, Long Ouyang, Daniel M Ziegler, Nisan Stiennon, Ryan Lowe, Jan Leike, and Paul Christiano. 2021a. Recursively summarizing books with human feedback. *arXiv preprint arXiv:2109.10862*.
- Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, and Wolfgang Macherey et al. 2016. Google’s neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.
- Zequ Wu, Michel Galley, Chris Brockett, Yizhe Zhang, and Bill Dolan. 2021b. Automatic document sketching: Generating drafts from analogous texts. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2102–2113, Online. Association for Computational Linguistics.
- Zequ Wu, Michel Galley, Chris Brockett, Yizhe Zhang, Xiang Gao, Chris Quirk, Rik Koncel-Kedziorski, Jianfeng Gao, Hannaneh Hajishirzi, Mari Ostendorf, and Bill Dolan. 2021c. A controllable model of grounded response generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 14085–14093.
- Zequ Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A. Smith, Mari Ostendorf, and Hannaneh Hajishirzi. 2023a. Fine-grained human feedback gives better rewards for language model training. In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Zequ Wu, Bo-Ru Lu, Hannaneh Hajishirzi, and Mari Ostendorf. 2021d. DIALKI: Knowledge identification in conversational systems through dialogue-document contextualization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1852–1863, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Zequ Wu, Yi Luan, Hannah Rashkin, David Reitter, Hannaneh Hajishirzi, Mari Ostendorf, and Gaurav Singh Tomar. 2022. CONQRR: Conversational query rewriting for retrieval with reinforcement learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

- Zequiu Wu, Ryu Parish, Hao Cheng, Sewon Min, Prithviraj Ammanabrolu, Mari Ostendorf, and Hannaneh Hajishirzi. 2023b. InSCIt: Information-seeking conversations with mixed-initiative interactions. *Transactions of the Association for Computational Linguistics*, 11:453–468.
- Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.
- Jing Xu, Megan Ung, Mojtaba Komeili, Kushal Arora, Y-Lan Boureau, and Jason Weston. 2023a. Learning new skills after deployment: Improving open-domain internet-driven dialogue with human feedback. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13557–13572, Toronto, Canada. Association for Computational Linguistics.
- Wenda Xu, Danqing Wang, Liangming Pan, Zhenqiao Song, Markus Freitag, William Wang, and Lei Li. 2023b. INSTRUCTSCORE: Towards explainable text generation evaluation with automatic feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5967–5994, Singapore. Association for Computational Linguistics.
- Peilin Yang, Hui Fang, and Jimmy Lin. 2017. Anserini: Enabling the use of lucene for information retrieval research. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '17*, page 1253–1256, New York, NY, USA. Association for Computing Machinery.
- Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with BERTserini. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 72–77, Minneapolis, Minnesota. Association for Computational Linguistics.
- Semih Yavuz, Abhinav Rastogi, Guan-Lin Chao, and Dilek Hakkani-Tur. 2019. DeepCopy: Grounded response generation with hierarchical pointer networks. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 122–132, Stockholm, Sweden. Association for Computational Linguistics.

- Shi Yu, Jiahua Liu, Jingqin Yang, Chenyan Xiong, Paul Bennett, Jianfeng Gao, and Zhiyuan Liu. 2020. Few-shot generative conversational query rewriting. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '20*, page 1933–1936, New York, NY, USA. Association for Computing Machinery.
- Shi Yu, Zhenghao Liu, Chenyan Xiong, Tao Feng, and Zhiyuan Liu. 2021. *Few-Shot Conversational Dense Retrieval*, page 829–838. Association for Computing Machinery, New York, NY, USA.
- Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. 2023. RRHF: Rank responses to align language models with human feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Xiang Yue, Boshi Wang, Ziru Chen, Kai Zhang, Yu Su, and Huan Sun. 2023. Automatic evaluation of attribution by large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4615–4635, Singapore. Association for Computational Linguistics.
- Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.
- Xueliang Zhao, Wei Wu, Can Xu, Chongyang Tao, Dongyan Zhao, and Rui Yan. 2020. Knowledge-grounded dialogue generation with pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3377–3390, Online. Association for Computational Linguistics.
- Zhou Zhao, Xinghua Jiang, Deng Cai, Jun Xiao, Xiaofei He, and Shiliang Pu. 2018. Multi-turn video question answering via multi-stream hierarchical attention context network. In *Proceedings of the Twenty-*

- Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 3690–3696. International Joint Conferences on Artificial Intelligence Organization.
- Chujie Zheng, Yunbo Cao, Daxin Jiang, and Minlie Huang. 2020. Difference-aware knowledge selection for knowledge-grounded conversation generation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 115–125, Online. Association for Computational Linguistics.
- Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018a. Common-sense knowledge aware conversation generation with graph attention. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4623–4629. International Joint Conferences on Artificial Intelligence Organization.
- Kangyan Zhou, Shrimai Prabhumoye, and Alan W Black. 2018b. A dataset for document grounded conversations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 708–713, Brussels, Belgium. Association for Computational Linguistics.
- Daniel M. Ziegler, Nisan Stiennon, Jeff Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *ArXiv*, abs/1909.08593.

Appendix A

Additional Examples and Training Details for FINE-GRAINED RLHF

A.1 Algorithm and Training Details of FINE-GRAINED RLHF

A.1.1 PPO with Fine-Grained Rewards

The algorithm below shows in detail how PPO updates the policy LM P_θ and the value model V_ψ with K fine-grained reward models R_{ϕ_k} .

A.1.2 Implementation Details

Model architectures. For the detoxification experiments, the policy model is initialized with GPT2-large [Radford et al., 2019], and the value model is initialized with GPT2-base. For the long-form QA experiments, the policy model is initialized with a supervised fine-tuned T5-large [Raffel et al., 2020], and the value model is initialized with T5-base. This design follows InstructGPT [Ouyang et al., 2022], which uses a larger (175B) policy model, and smaller value and reward (6B) models.

Training details on detoxification. For both the holistic reward baseline and the sentence-level (fine-grained) reward, we do a hyper-parameter search with the same set of hyper-parameters. For training, we run 200K episodes. The batch size (number of episodes per card during training) is 64. We use Adam optimizer with a linear learning rate scheduler and 10 warmup steps. We perform a hyper-parameter grid-

Algorithm 1 Fine-Grained Reinforcement Learning from Human Feedback (FINE-GRAINED RLHF)

Input initial policy model $P_{\theta_{\text{init}}}$; initial value model $V_{\psi_{\text{init}}}$; K reward models R_{ϕ_k} trained from human feedback; task prompts \mathcal{D} ; hyperparameters $\gamma, \lambda, \epsilon, \beta$

- 1: policy model $P_{\theta} \leftarrow P_{\theta_{\text{init}}}$, value model $V_{\psi} \leftarrow V_{\psi_{\text{init}}}$
- 2: **for** step = 1, ..., M **do**
- 3: Sample a batch \mathcal{D}_b from \mathcal{D}
- 4: Sample output sequence $y^n \sim P_{\theta}(\cdot | x^n)$ for each prompt $x^n \in \mathcal{D}_b$
- 5: Compute rewards $\{r_t^n\}_{t=1}^{|y^n|}$ for each sampled output y^n by running R_{ϕ_k} ▷ Eq. 6.1
- 6: Compute advantages $\{A_t\}_{t=1}^{|y^n|}$ and value targets $\{V^{\text{targ}}(s_t)\}_{t=1}^{|y^n|}$ for each y^n with V_{ψ}
- 7: **for** PPO iteration = 1, ..., μ **do**
- 8: Update the policy model by maximizing the PPO clipped surrogate objective:

$$\theta \leftarrow \arg \max_{\theta} \frac{1}{|\mathcal{D}_b|} \sum_{n=1}^{|\mathcal{D}_b|} \frac{1}{|y^n|} \sum_{t=1}^{|y^n|} \min \left(\frac{P_{\theta}(a_t | s_t)}{P_{\theta_{\text{old}}}(a_t | s_t)} A_t, \text{clip}(v_t, 1 - \epsilon, 1 + \epsilon) A_t \right)$$

- 9: Update the value model by minimizing a square-error objective:

$$\psi \leftarrow \arg \min_{\psi} \frac{1}{|\mathcal{D}_b|} \sum_{n=1}^{|\mathcal{D}_b|} \frac{1}{|y^n|} \sum_{t=1}^{|y^n|} (V_{\psi}(s_t) - V^{\text{targ}}(s_t))^2$$

Output P_{θ}

search for peak learning rate $\in \{5e-6, 1e-5, 2e-5\}$, KL coefficient $\beta \in \{0.1, 0.2, 0.3\}$, discounting factor $\lambda \in \{0.95, 0.97, 0.99\}$, and the frequency of exploration (number of sampled outputs) $\in \{2, 4, 8\}$. We find that the higher the KL coefficient, the lower the perplexity, and the higher toxicity. This is consistent with findings from previous RLHF studies [Ouyang et al., 2022; Ramamurthy et al., 2023]. For a fair comparison, we eventually choose a set of parameters that achieve a similar level of perplexity for both reward models. The optimal set of hyper-parameters for holistic reward is $\beta = 0.3, \lambda = 0.99$. For sentence-level reward $\beta = 0.1, \lambda = 0.95$. The learning rate is $1e-5$, and the exploration frequency is 4 for both experiments. We choose the checkpoint with the lowest validation set toxicity for evaluation. Regarding computation time, we use $2 \times 80\text{G}$ NVIDIA A100 GPU for training, and the run time is about 22 hours.

Training details on long-form QA. We conduct a similar hyper-parameter grid search as our detoxification experiments. For long-Form QA, the input length limit is 1024, and the output length limit is 200. Notice that this is much longer than detoxification, so we use a smaller batch size and fewer training episodes. We experiment with multiple combinations of reward model weights. Fixing $w_2 = 0.5$ (factuality reward weight), we perform a grid search on $w_1, w_3 \in [0.0, 0.5]$. We eventually choose $w_1 = 0.3, w_2 = 0.5, w_3 =$

0.3, which reaches a balance between three reward models and allows all three rewards to increase during training. For training, the batch size (number of episodes per card during training) is 32. We use Adam optimizer with a linear learning rate scheduler and 100 warmup steps. We perform a hyper-parameter grid-search for peak learning rate $\in \{5e-6, 1e-5, 2e-5\}$, KL coefficient $\beta \in \{0.1, 0.2, 0.3\}$, discounting factor $\lambda \in \{0.95, 0.97, 0.99\}$, and the frequency of exploration $\in \{2, 4, 8\}$. The optimal set of hyper-parameters for Pref. RLHF is $\beta = 0.2, \lambda = 0.99$. For FINE-GRAINED RLHF, $\beta = 0.3, \lambda = 0.95$. The learning rate is $1e-5$, and the exploration frequency is 4 for both experiments. we run 80K episodes, which is approximately 5 epochs. We choose the checkpoint with the highest validation reward for evaluation. Regarding computation time, we use $2 \times 80\text{G}$ NVIDIA A100 GPU for training, and the run time is about 15 hours.

A note on the error bars. All results we report in the chapter are from 3 independent runs. The scores reported are all averaged across all runs. The error bars are represented as the shades behind each training curve in our figures. It shows the standard error across three runs.

A.2 Long-Form QA Data and Human Feedback Annotation

A.2.1 Human Feedback Annotation

Fine-grained feedback. As discussed in § 6.4.1, we first use 1K randomly sampled training examples to train a T5-large based supervised model **SFT** as the initial policy model $P_{\theta_{init}}$. Then we collect feedback on sampled outputs from SFT for the remaining 2,853 training examples and the 500 development examples, using the Amazon Mechanical Turk platform.¹

Figure A.1 shows the fine-grained human feedback annotation interface with an example from QA-FEEDBACK. In addition to the task input—question q and oracle passages P , we also provide a human-written response from ASQA to the worker as reference. However, it is important to note that, in practice, the annotation of our fine-grained feedback should not require the human-written response. The only purpose for us to provide the gold response is to have our workers follow the same question interpretation and expected response of the workers who annotate for ASQA, such that our experimental comparison with supervised

¹<https://www.mturk.com/>

models (**SFT** and **SFT-Full**; details in § 6.4.3) is fair. However, we still instruct our workers to strictly use the given passages for checking factual errors. For each span error, we ask the worker to select one out of 5 categories shown in Figure A.2 (left).² However, we collapse these 5 categories into two categories (C_1 and C_2 mentioned in § 6.4.1) based on whether the error detection depends on the passages or not. When workers mark passage sentences as containing missing information, we instruct them to categorize each sentence as missing “answer”, “major auxiliary information” or “minor auxiliary information,” as shown in Figure A.2 (right). Our instruction to the worker is provided in Figure A.4.

Quality control. Before feedback collection, we design a qualification task to select qualified workers for this feedback annotation task. The qualification task consists of 5 questions with their corresponding passages and model outputs for the workers to annotate. We manually review about 70 submissions of the qualification task and select 15 workers whose annotation is marked by us as of high quality. Throughout the actual feedback annotation process, we constantly monitor the annotated data and send feedback to workers.

Preference-based feedback. For comparison purposes, we follow Ouyang et al. [2022] to collect pairwise human preferences from the same group of workers we select from the qualification task. We sample four model outputs for each prompt x , which gives 6 pairs of model outputs. Similarly, we provide the worker with the human-written response and ask the workers to indicate pairwise preferences (ties are allowed) based on all errors they can find each model output. Figure A.3 shows the preference-based human feedback annotation interface with an example from QA-FEEDBACK.

Pay structure. We pay a base rate of \$1.5 per example for annotating fine-grained or preference feedback. If the example consists of ≥ 3 passages to read, we assign an additional \$0.3 bonus to the example. On average, we pay roughly \$1.65 per example for both tasks, which gives an \$16.5 hourly pay for our workers.

A.2.2 Analysis of Collected Fine-Grained Feedback

Overall, among all *error spans* we collect, 76% of them are C_1 errors and the remaining 24% are C_2 errors. However, it is important to note that we instruct workers to label C_2 errors only at places that don’t have a C_1 error. 75% examples are labeled as being incomplete; i.e., containing missing information that can be found in the given passages (C_3). Among all marked passage sentences that contain missing information, 31%,

²We see very few “incoherence” errors (1%), so the majority of labeled errors are from the other four categories during annotation.

42% and 27% are missing answer, major auxiliary information and minor auxiliary information respectively.

To analyze human-human agreement, a subset of 300 examples receive annotations from two distinct workers. We observe that while the exact agreement in error span boundaries is low, workers achieve reasonably high agreement on whether a sub-sentence contains C_1 (reach an agreement for 83% of all sub-sentences) and whether a sentence contains C_2 (92%).³ The agreement on whether a model output contains complete information or not (C_3) is 85%. Therefore, we decide to have the density for error type C_1 , C_2 , and C_3 as sub-sentence, sentence and full sequence.

A.3 Long-Form QA Reward Model Training Details

We train reward models with the 2,835 training examples with feedback collected and select the best model for each error category based on their performance on the development set. The batch size and training epochs are 24 and 50 for R_{ϕ_1} and R_{ϕ_2} . Each training is run on a single 80G NVIDIA A100 GPU, taking 1 and 2 hours for training R_{ϕ_1} and R_{ϕ_2} respectively.⁴ The batch size and training epochs are 12 (per GPU) and 30 for R_{ϕ_3} and the preference-based reward model. Each training is run on $2 \times 80G$ NVIDIA A100 GPU and takes 2 hours. We use Adam optimizer with a linear learning rate scheduler for all reward model training. For each reward model, we search the learning rate over $\{5e^{-6}, 1e^{-5}, 5e^{-5}\}$, weight decay over $\{0.001, 0.01\}$, and warm-up step ratio over $\{0.1, 0.2\}$ based on the dev set performance. Specifically, we use the model for R_{ϕ_1} and R_{ϕ_2} that achieve the best binary classification accuracy. For R_{ϕ_3} and the preference-based reward model, we select the model that achieves the best pairwise comparison accuracy. We also provide more training details for each reward model below.

R_{ϕ_1} for C_1 : Irrelevance, repetition, or incoherence. To train the reward model R_{ϕ_1} that detects error of irrelevance, repetition, or incoherence, we apply a token-level classification loss to each `[sep]` token before y_j^1 , where its gold label g_j is “has error” if there is a $f_i \in \mathcal{F}$ that has a_{b_i, \dots, e_i} overlapped with y_j^1 and $c_i = 1$, and “no error” otherwise. We observe that most of the spans marked as error type C_1 that are shorter than 5 words usually carry very little information or are annotated as a result of workers being very careful or strict. Therefore, we filter out such short spans before constructing training examples

³We use spaCy [Honnibal et al., 2020] to segment generated model outputs into sentences. We then split sentences into sub-sentences using a comma or semicolon.

⁴Note that training R_{ϕ_1} takes shorter time as its input does not contain passages.

for R_{ϕ_1} . Overall, we get 7379 and 8059 sub-sentences with the “has error” and “no error” label respectively.

R_{ϕ_2} for C_2 : Incorrect or unverifiable facts. We train R_{ϕ_2} in a similar way as how we train R_{ϕ_1} . Instead of predicting the error for each sub-sentence, R_{ϕ_2} is trained to predict at the sentence level (i.e., y_j^2 is the j^{th} sentence in y). Since workers do not annotate C_2 error for spans that are already labeled as having C_1 error, in order to avoid false negatives in training R_{ϕ_2} , we do not provide gold label nor calculate loss for sentences that only contain C_1 error from training. In other words, all sentences that contain a C_2 error has the gold label “has error” and sentences that contain no error has the gold label “no error”. Overall, we get 1600 and 3411 sentences with the “has error” and “no error” label respectively.

R_{ϕ_3} for C_3 : Incomplete information. Instead of casting this as a classification task, R_{ϕ_3} predicts a single scalar reward given x and y and is trained with a pairwise comparison loss [Ouyang et al., 2022]. This is motivated by early work [Li et al., 2019a] that shows the better reliability of pairwise comparison than error classification when assessing a full generation sequence. To construct training data for R_{ϕ_3} , we bootstrap pairwise comparisons from the corrected model output y' as follows. We first map each sub-sentence in y' to a passage sentence in P that has a sub-string with the highest token-level F1 score with the sub-sentence,⁵ and denote all mapped sentences as S . We then sample four responses from SFT, for each we do the same sentence mapping to get a set of passages sentences S' . We calculate $score(y) = |S' \cap S|/|S|$ as the information completeness score for each model response y . We follow Ouyang et al. [2022] to pair up sampled responses for q and denote each sampled response pair as (\bar{y}_p, \bar{y}_l) , where $score(\bar{y}_p) > score(\bar{y}_l)$. We drop the pairs where $score(\bar{y}_p) = score(\bar{y}_l)$. Then we follow Ouyang et al. [2022] to train R_{ϕ_3} with the loss function in Eq. 6.2. We have a total number of 6821 pair examples in training.

Preference-based reward model. The preference-based reward model is trained in a similar way as R_{ϕ_3} , with \bar{y}_p representing the human preferred response against \bar{y}_l in the loss function Eq. 6.2. We drop the pairs where a tie is indicated. We have a total number of 14981 pair examples in training.

⁵We manually review 50 mapped passage sentences and find over 90% of them are correctly mapped, which indicates frequent extractive behaviors from $P_{\theta_{init}}$.

Input Question:
When was the parliament formed in the uk?

Passage 1: Title (So) - Parliament of the United Kingdom [Click to hide](#)

S1. The Royal and Parliamentary Titles Act 1927 formally amended the name to the "Parliament of the United Kingdom of Great Britain and Northern Ireland", five years after the secession of the Irish Free State in 1922.
S2. With the global expansion of the British Empire, the UK Parliament has shaped the political systems of many countries as ex-colonies and so it has been called the "Mother of Parliaments".
S3. However, John Bright who coined the epithet used it in reference to the political culture of "England" rather than just the parliamentary system.
S4. In theory, the UK's supreme legislative power is officially vested in the Crown-in-Parliament.
S5. However, the Crown normally acts on the advice of the Prime Minister and the powers of the House of Lords are limited to only delaying legislation; thus power is "de facto" vested in the House of Commons.
S6. The United Kingdom of Great Britain and Ireland was created on 1 January 1801, by the merger of the Kingdoms of Great Britain and Ireland under the Acts of Union 1800.
S7. The principle of ministerial responsibility to the lower House did not develop until the 19th century—the House of Lords was superior to the House of Commons both in theory and in practice.
S8. Members of the House of Commons (MPs) were elected in an antiquated electoral system, under which constituencies of vastly different sizes existed.

Reference Response:
The parliament formed in the UK was initially called Parliament of the United Kingdom of Great Britain and Ireland but the Royal and Parliamentary Titles Act 1927 formally amended the name to the "Parliament of the United Kingdom of Great Britain and Northern Ireland", five years after the secession of the Irish Free State. Parliament of the United Kingdom of Great Britain and Ireland was created on 1 January 1801, while the parliament formed in the United Kingdom of Great Britain and Northern Ireland was created on 11 November 1920.

STEP 1: Error labeling for the model-predicted response: [Click to hide](#) *** Hint: The mistake labeling prompt window is draggable. ***

After **CAREFUL** checking, I find no error in this predicted response!

The United Kingdom of Great Britain and Ireland was created on 1 January 1801, by the merger of the Kingdoms of Great Britain and Ireland under the Acts of Union 1800. On 1 January 1801 the parliament was formed in London.

✗ Unverifiable **✗ Irrelevant**

[Add Missing Info](#) *** Hint: If the error type selection doesn't work, try double clicking. ***

✗ Missing Major Auxiliary Info: [explanation] {"passage_id":1,"sentence_id":1}

STEP 2: Your correction of the model-predicted response: [Click to hide](#)

The Royal and Parliamentary Titles Act 1927 formally amended the name to the "Parliament of the United Kingdom of Great Britain and Northern Ireland", five years after the secession of the Irish Free State. Parliament of the United Kingdom of Great Britain and Ireland was created on 1 January 1801.

[Reset](#)

Figure A.1: Fine-grained feedback annotation interface.

A.4 Qualitative Examples for Long-Form QA

A.4.1 Examples on LM Customization

As discussed in § 6.4.5, we can modify the weight of each fine-grained reward model during RL training to get LM with different behaviors. Here, we explore three configurations of reward model weights and name them ‘short’, ‘medium’, and ‘long’ based on the LM’s average generation length. The ‘short’ configuration generates concise and short responses, while the ‘long’ configuration generates detailed and long responses. Table A.1 demonstrates the different behaviors of our customized LMs. Given the same question, each LM generates different amount of auxiliary information in the response.

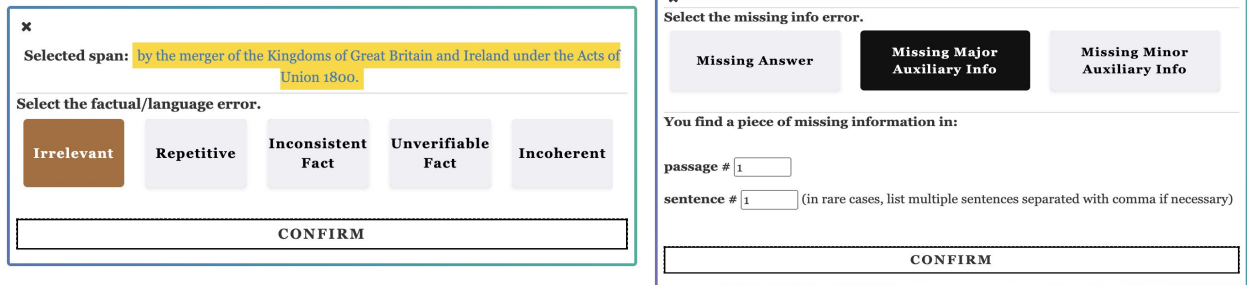


Figure A.2: Error types in the fine-grained feedback annotation interface.

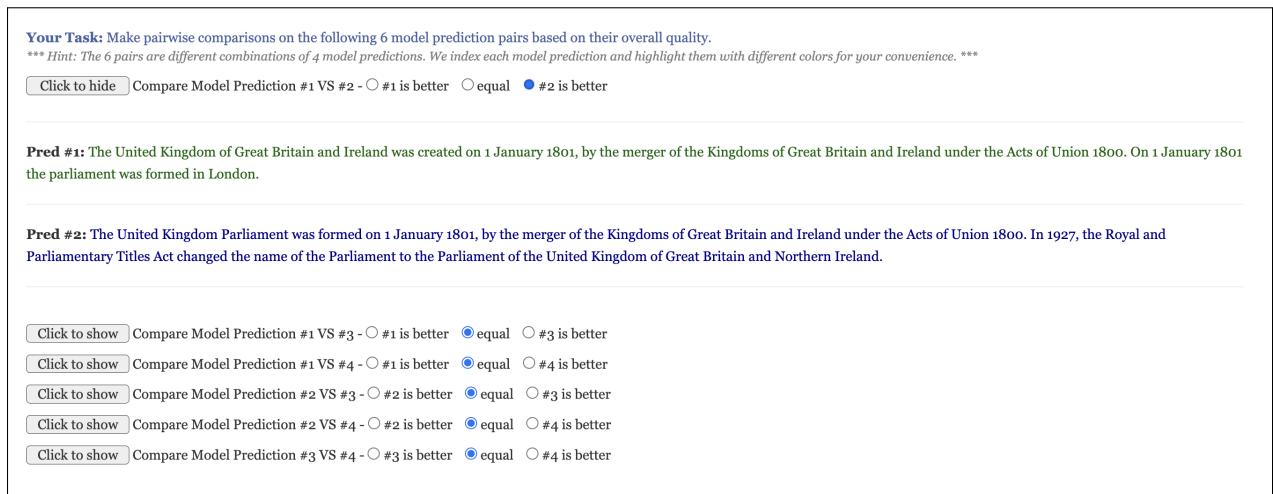


Figure A.3: Preference feedback annotation interface. The task input is omitted (same as in Figure A.1).

In each task, **you will be given** a question, a set of Wikipedia passages (with their article title provided), a reference response, and a model-predicted response. Each passage is presented as a sequence of sentences (title is indexed as sentence #0). **Your goal is to** 1) label mistakes made in the model prediction and 2) make corrective edits to the model prediction based on your labeled mistakes.

Important Definitions: An *ideal response* to a question should provide both **answer(s)** that directly responds to the question and **crucial auxiliary information** for better comprehension of the answer(s). We consider auxiliary information as *crucial* if it is used in the reference response. Additionally, all information in an *ideal response* should be **factually consistent with (i.e., grounded in) the passages**. Note that the reference response is written by a human with potentially different grounding passages, and thus, you might find **answers** that can be found in the passages but are not included in the reference, which are **STILL** expected to be in the model prediction. On the other hand, answers in the reference that cannot be found in or verifiable by the passages are **NOT** expected to be in the model prediction. **To conclude, all answers** are expected in the model prediction **IF AND ONLY IF** it can be found in the passages. **Crucial auxiliary information** is expected in the model prediction **IF AND ONLY IF** it can be found in both the reference response and the passages.

Here are the detailed annotation steps:

STEP 1 - Read the question and label mistakes made in a model-predicted response. As explained above, leverage the reference, **BUT** rely on the passages. Decide the mistake type and follow detailed instructions as follows. We encourage you to use CTRL+F/CMD+F for navigating the reference and passages using keywords.

- **Erroneous Span (i.e., substring):** Highlight each span in the model prediction that contains one of the following errors. Label each span **as short as possible** and make sure each labeled span only contain **one single** information piece. You will be prompted to select the error type and provide an explanation if apply. For each span, label **only one** error type. If a span contains multiple errors, select the error type based on the order below (e.g., if a span contains "irrelevant" information, ignore any "inconsistent fact" it contains).
 - i. **[Irrelevant]:** The span contains "irrelevant" information (e.g., neither an answer nor crucial auxiliary information, defined in the first 2 sentences in "*Important Definitions*"). To detect "irrelevant" errors, you do not need to consider whether the information is factually consistent with the passages or not.
 - ii. **[Repetitive]:** The span repeats information in its previous text. Provide the previous text being repeated (**as short as possible**). **Note that** a "repetitive" span should still be marked even if its previous text being repeated contains an factual or coherence issue (defined below). However, if the previous text is "irrelevant", it should be marked as "irrelevant" too.
 - iii. **[Inconsistent Fact]:** The span is factually inconsistent with the passages. Enter the passage id and sentence id(s) as evidence. **Note that** if you find multiple evidences in the passages, mark only one of them. The need for multiple passage ids usually indicates that you should separate the error into multiple ones (due to multiple information pieces).
 - iv. **[Unverifiable Fact]:** The span is factually unverifiable (i.e., not mentioned in any passage), after **carefully** checking all passages. Common sense (e.g., "a bicycle has two wheels") doesn't need to be verified. However, do not count knowledge only commonly known in a specific region/community as commonsense. This can be subjective, and simply follow your best judgment.
 - v. **[Incoherent]:** The span contains major grammar error (ignore minor typos), is uninterpretable, contradicts to common sense, or is not coherent with its context.
- **Missing Information:** Identify information that is expected but missing in the model prediction. Check "*Important Definitions*" above to see how to identify such information. Classify **each piece** of missing information as **[Missing Answer]** or **[Missing Major/Minor Auxiliary Information]**, and enter the passage id and sentence id(s) as evidence. Mark the missing auxiliary information as **major** if you think the information is indeed helpful for understanding the answer. Otherwise (e.g., a bit off-topic), mark it as **minor**. Simply follow your best judgment. Follow the same "**Note that**" rule under "*Inconsistent Fact*" above.

STEP 2 - Correct the model prediction to address your labeled mistakes in STEP 1. **Instead of** copying and pasting the reference into the box, make minimal local edits on the original prediction. Make sure the corrected version is fluent and free of typos. In rare cases when you find no useful information in passages for answering the question, you can leave the box as blank if necessary.

Figure A.4: Fine-grained feedback annotation instructions.

Question:	When did the French join revolution on colonists' side?
Passages:	<p>Article Title: France in the American Revolutionary War French involvement in the American Revolutionary War began in 1775, when France, a rival of the British Empire, secretly shipped supplies to the Continental Army. A Treaty of Alliance in 1778 soon followed, which led to shipments of money and material to the United States. Subsequently, the Spanish Empire and the Dutch Republic also began to send assistance, leaving the British Empire with no allies. France's help is considered a vital and decisive contribution to the United States' victory against the British. As a cost of participation in the war, France accumulated over 1 billion livres in debt. After its defeat in the Seven Years' War in 1763, France lost its vast holdings in North America. Meanwhile, the American colonists and the British government began to fight over whether Parliament in London or the colonial assemblies had primary responsibility for taxation. As part of that conflict, the colonists organized the Boston Tea Party in response to a tax on tea. The British government responded by passing the Intolerable Acts, which included the closing of Boston Harbor and the revocation of Massachusetts's colonial charter. This conflict exacerbated tensions further. The ideological conflict escalated into open warfare in 1775, at which point the American patriots revolted against British rule. France, who had been rebuilding their Navy and other forces, saw this as an opportunity to seriously weaken her perennial enemy.</p> <p>Article Title: France in the American Revolutionary War France bitterly resented its loss in the Seven Years' War and sought revenge. It also wanted to strategically weaken Britain. Following the Declaration of Independence, the American Revolution was well received by both the general population and the aristocracy in France. The Revolution was perceived as the incarnation of the Enlightenment Spirit against the "English tyranny." Benjamin Franklin traveled to France in December 1776 in order to rally the nation's support, and he was welcomed with great enthusiasm. At first, French support was covert: French agents sent the Patriots military aid (predominantly gunpowder) through a company called "Rodrigue Hortalez et Compagnie", beginning in the spring of 1776. Estimates place the percentage of French supplied arms to the Americans in the Saratoga campaign at up to 90%. By 1777, over five million "livres" of aid had been sent to the American rebels. Motivated by the prospect of glory in battle or animated by the sincere ideals of liberty and republicanism, volunteers like Pierre Charles L'Enfant joined the American army. The most famous was Lafayette, a charming young aristocrat who defied the king's order and enlisted in 1777 at age 20. He became an aide to George Washington and a combat general. More importantly, he solidified a favorable American view of France.</p> <p>Article Title: Treaty of Alliance (1778) On March 17, 1778, four days after a French ambassador informed the British government that France had officially recognized the United States as an independent nation with the signing of the Treaty of Amity and Commerce and the Treaty of Alliance, Great Britain declared war on France, thereby engaging that nation in the American Revolutionary War. French entry into the war would lead to further escalation of the war when Spain entered the fight against Britain as France's ally, after the signing of the Treaty of Aranjuez on April 12, 1779, and again in December 1780 when Britain declared war on the Dutch Republic after seizing a Dutch merchant ship they claimed was carrying contraband to France during the Affair of Fielding and Bylandt. After the signing of the treaty French supplies of arms, ammunition, and uniforms proved vital for the Continental Army, while their actions in the Anglo French War of 1778-83 in the West Indies and elsewhere forced Britain to redeploy troops and naval units away from the North American colonies to secure their holdings.</p>
LM outputs with different reward model configurations:	
Short:	French involvement in the American Revolutionary War began in 1775, when France, a rival of the British Empire, secretly shipped supplies to the Continental Army. A Treaty of Alliance in 1778 soon followed, which led to shipments of money and matériel to the United States. Subsequently, the Spanish Empire and the Dutch Republic also began to send assistance, leaving the British Empire with no allies.
Medium:	French involvement in the American Revolutionary War began in 1775, when France, a rival of the British Empire, secretly shipped supplies to the Continental Army. A Treaty of Alliance in 1778 soon followed, which led to shipments of money and matériel to the United States. Subsequently, the Spanish Empire and the Dutch Republic also began to send assistance, leaving the British Empire with no allies. On March 17, 1778, four days after a French ambassador informed the British government that France had officially recognized the United States as an independent nation, Great Britain declared war on France, thereby engaging that nation in the American Revolutionary War.
Long:	French involvement in the American Revolutionary War began in 1775, when France, a rival of the British Empire, secretly shipped supplies to the Continental Army. A Treaty of Alliance in 1778 soon followed, which led to shipments of money and matériel to the United States. Subsequently, the Spanish Empire and the Dutch Republic also began to send assistance, leaving the British Empire with no allies. On March 17, 1778, four days after a French ambassador informed the British government that France had officially recognized the United States as an independent nation with the signing of the Treaty of Amity and Commerce and the Treaty of Alliance, Great Britain declared war on France, thereby engaging that nation in the American Revolutionary War.

Table A.1: Qualitative examples on LM customization.

A.4.2 Examples on LM Errors

Table A.2 and Table A.3 show examples of LM outputs from all the compared systems (SFT, Pref. RLHF, and FINE-GRAINED RLHF). We mark the fine-grained errors on the model outputs. Overall, our FINE-GRAINED RLHF outperforms SFT and Pref. RLHF in all three error types.

Question:	Who was the prime minister of Canada in 1920?
Passages:	<p>Article Title: Robert Borden Not only did Borden’s persistence allow him to represent Canada in Paris as a nation, it also ensured that each of the dominions could sign the Treaty of Versailles in its own right, and receive a separate membership in the League of Nations. During the conference Borden tried to act as an intermediary between the United States and other members of the British Empire delegation, particularly Australia and New Zealand over the issue of Mandates. Borden also discussed with Lloyd George, the possibility of Canada taking over the administration of Belize and the West Indies, but no agreement was reached. At Borden’s insistence, the treaty was ratified by the Canadian Parliament. Borden was the last Prime Minister to be knighted after the House of Commons indicated its desire for the discontinuation of the granting of any future titles to Canadians in 1919 with the adoption of the Nickle Resolution. In 1919 Borden approved the use of troops to put down the Winnipeg general strike, which was feared to be the result of Bolshevik agitation from the Soviet Union. Sir Robert Borden retired from office in 1920. He was the Chancellor of Queen’s University from 1924 to 1930 and also was Chancellor of McGill University from 1918 to 1920 while still Prime Minister. Borden also served as Vice-President of The Champlain Society between 1923 and 1925. He was the Society’s first Honorary President between 1925 and 1938.</p> <p>Article Title: Robert Borden Sir Robert Laird Borden, (June 26, 1854 – June 10, 1937) was a Canadian lawyer and politician who served as the eighth Prime Minister of Canada, in office from 1911 to 1920. He is best known for his leadership of Canada during World War I. Borden was born in Grand-Pré, Nova Scotia. He worked as a schoolteacher for a period and then served his articles of clerkship at a Halifax law firm. He was called to the bar in 1878, and soon became one of Nova Scotia’s most prominent barristers. Borden was elected to the House of Commons of Canada in 1896, representing the Conservative Party. He replaced Charles Tupper as party leader in 1901, and became prime minister after the party’s victory at the 1911 federal election. As prime minister, Borden led Canada through World War I and its immediate aftermath. His government passed the "War Measures Act", created the Canadian Expeditionary Force, and eventually introduced compulsory military service, which sparked the 1917 conscription crisis. On the home front, it dealt with the consequences of the Halifax Explosion, introduced women’s suffrage for federal elections, and used the North-West Mounted Police to break up the 1919 Winnipeg general strike.</p> <p>Article Title: Arthur Meighen Arthur Meighen Arthur Meighen (; 16 June 1874 – 5 August 1960) was a Canadian lawyer and politician who served as the ninth Prime Minister of Canada, in office from July 1920 to December 1921 and again from June to September 1926. He led the Conservative Party from 1920 to 1926 and from 1941 to 1942. Meighen was born in rural Perth County, Ontario. He studied mathematics at the University of Toronto, and then went on to Osgoode Hall Law School. After qualifying to practice law, he moved to Portage la Prairie, Manitoba. Meighen entered the House of Commons of Canada in 1908, aged 34, and in 1913 was appointed to the Cabinet of Prime Minister Sir Robert Borden. Under Borden, Meighen served as Solicitor General (1913–1917), Secretary of State for Canada (1917), Minister of Mines (1917; 1919–1920), Minister of the Interior (1917–1920), and Superintendent-General of Indian Affairs (1917–1920). In 1920, Meighen succeeded Borden as Conservative leader and Prime Minister – the first born after Confederation, and the only one from a Manitoba riding. He suffered a heavy defeat in the 1921 election to Mackenzie King and the Liberal Party, but re-entered Parliament through a 1922 by-election and remained as Opposition Leader.</p>
LM outputs	
SFT:	Sir Robert Borden was the eighth Prime Minister of Canada from 1911 to 1920. Arthur Meighen was the ninth Prime Minister of Canada from 1920 to 1926.
Pref. RLHF:	Sir Robert Laird Borden was a Canadian lawyer and politician who served as the eighth Prime Minister of Canada, in office from 1911 to 1920. Arthur Meighen was a Canadian lawyer and politician who served as the ninth Prime Minister of Canada, in office from July 1920 to December 1921 and again from June to September 1926. He led the Conservative Party from 1920 to 1926 and from 1941 to 1942.
FINE-GRAINED RLHF	Sir Robert Laird Borden was a Canadian lawyer and politician who served as the eighth Prime Minister of Canada, in office from 1911 to 1920. Arthur Meighen was a Canadian lawyer and politician who served as the ninth Prime Minister of Canada, in office from July 1920 to December 1921 and again from June to September 1926.

Table A.2: Examples of errors in LM outputs of SFT, Pref. RLHF, and our FINE-GRAINED RLHF. We mark the C_1 (irrelevance, repetition, or incoherence) and C_2 (incorrect or unverifiable facts) types of errors in the model outputs.

Question:	Where does the flat iron cut of meat come from?
Passages:	<p>Article Title: Flat iron steak</p> <p>Flat iron steak (US), butlers' steak (UK), or oyster blade steak (Australia and New Zealand) is a cut of steak cut with the grain from the shoulder of the animal. This produces a flavorful cut that is a bit tough because it contains a gristly fascia membrane unless removed. Some restaurants offer it on their menu, often at lower price than the more popular rib-eye and strip steaks of the same grade. This is used, in some places, as a means of selling a less expensive cut from the same animal, for example Kobe beef. This cut of steak is from the shoulder of a beef animal. It is located adjacent to the heart of the shoulder clod, under the seven or paddle bone (shoulder blade or scapula). The steak encompasses the infraspinatus muscles of beef, and one may see this displayed in some butcher shops and meat markets as a "top blade" roast. Anatomically, the muscle forms the dorsal part of the rotator cuff of the steer. This cut is anatomically distinct from the shoulder tender, which lies directly below it and is the teres major. Flat iron steaks usually have a significant amount of marbling.</p>
LM outputs	
SFT:	Flat iron steak is a cut of steak cut with the grain from the shoulder of the animal. This produces a flavorful cut that is a bit tough because it contains a gristly fascia membrane unless removed. Some restaurants offer it on their menu, often at lower price than the more popular rib-eye and strip steaks of the same grade. This is used, in some places, as a means of selling a less expensive cut from the same animal, for example Kobe beef.
Pref. RLHF:	Flat iron steak is a cut of steak cut with the grain from the shoulder of the animal. This produces a flavorful cut that is a bit tough because it contains a gristly fascia membrane unless removed. Some restaurants offer it on their menu, often at lower price than the more popular rib-eye and strip steaks of the same grade. This is used, in some places, as a means of selling a less expensive cut from the same animal, for example Kobe beef.
FINE-GRAINED RLHF	Flat iron steak is a cut of steak cut with the grain from the shoulder of the animal. This is located adjacent to the heart of the shoulder clod, under the seven or paddle bone (shoulder blade or scapula). This steak encompasses the infraspinatus muscles of beef, and one may see this displayed in some butcher shops and meat markets as a "top blade roast."

Table A.3: Qualitative examples on another question. Here we highlight the information in blue that is covered by the LM trained with FINE-GRAINED RLHF but missed (C_3 : incomplete information error) by SFT and Pref.RLHF.

A.4.3 Examples on Reward Model Ablation

As discussed in § 6.4.6, reward models are competing against each other, and we experiment with removing one of the three reward models during RL training. Table A.4 shows an example of how LMs behave in such scenarios. See § 6.4.6 for our observations.

Question:	How much does it cost to raise the Costa Concordia?
Passages:	<p>Article Title: Costa Pacifica Costa Pacifica is a "Concordia"-class cruise ship for Costa Crociere. She was handed over to Costa Crociere on 29 May 2009. Her sister ships, "Costa Concordia" and "Costa Serena", were launched in 2006 and in 2007, with "Costa Favolosa" and "Costa Fascinosa" launched in 2011 and 2012 respectively. Onboard facilities include the Samsara Spa and PlayStation World, an area boasting PS3 consoles, and the first recording studio at sea. "Costa Pacifica" is the third ship of the "Concordia"-class, preceded by sister ships "Costa Concordia" and "Costa Serena" and was followed by "Costa Favolosa" and "Costa Fascinosa" on 2011 and 2012, all part of the expansion program of Costa which entailed an investment of 2.4 billion Euro, and is currently the largest fleet expansion program in the world. "Costa Pacifica" was ordered on 14 December 2005, by Carnival Corporation, the parent company of Costa Crociere. The order was for four ships, which includes; "Carnival Dream", for Carnival Cruise Lines; "Nieuw Amsterdam" for Holland America Line; and "Ruby Princess" for Princess Cruises, at a total cost of US \$3 billion.</p> <p>Article Title: Costa Concordia disaster On 13 January 2012, the Italian cruise ship "Costa Concordia" ran aground and overturned after striking an underwater rock off Isola del Giglio, Tuscany, resulting in 32 deaths. The eight year old Costa Cruises vessel was on the first leg of a cruise around the Mediterranean Sea when she deviated from her planned route at the Isola del Giglio, sailed closer to the island, and struck a rock formation on the sea floor. A six-hour rescue effort resulted in most of the passengers being brought ashore. "Costa Concordia" was officially declared a "constructive total loss" by the insurance company, and her salvage was "one of the biggest maritime salvage operations". On 16 September 2013, the parbuckle salvage of the ship began, and by the early hours of 17 September 2013, the ship was set upright on its underwater cradle. In July 2014, the ship was refloated by large sponsons (metal tanks) welded to its sides and was towed 320 kilometres (200 miles) to its home port of Genoa for scrapping which was finished in July 2017. The total cost of the disaster, including victims' compensation, refloating, towing and scrapping costs, is estimated at approximately \$2 billion, more than three times the \$612 million construction cost of the ship.</p>
LM outputs: removing one fine-grained reward model during training	
All reward models:	Costa Concordia is a cruise ship for Costa Crociere. On 13 January 2012, the Italian cruise ship Costa Concordia ran aground and overturned after striking an underwater rock off Isola del Giglio, Tuscany, resulting in 32 deaths. The total cost of the disaster, including victims' compensation, refloating, towing and scrapping costs, is estimated at approximately \$2 billion, more than three times the \$612 million construction cost of the ship.
Without relevance RM R_{ϕ_1}:	<p>On 13 January 2012, the Italian cruise ship Costa Concordia ran aground and overturned after striking an underwater rock off Isola del Giglio, Tuscany, resulting in 32 deaths. On 16 September 2013, the parbuckle salvage of the ship began, and by the early hours of 17 September 2013, the ship was set upright on its underwater cradle. In July 2014, the ship was refloated by large sponsons (metal tanks welded to its sides and was towed 320 kilometres (200 miles) to its home port of Genoa for scrapping which was finished in July 2017. The total cost of the disaster, including victims' compensation, refloating, refloating, towing and scrapping costs, is estimated at approximately \$2 billion, more than three times the \$612 million construction cost of the ship. The total cost of the disaster, including victims' compensation, refloating, towing and scrapping costs, is estimated at approximately \$2 billion.</p> <p>Comment: Too much irrelevant information & repetition error.</p>
Without factuality RM R_{ϕ_2}:	<p>The cost to raise the Costa Concordia was estimated at approximately \$2 billion, more than three times the \$612 million construction cost of the ship.</p> <p>Comment: Factuality error. Direct answer without any supporting evidence from the passages.</p>
Without info. completeness RM R_{ϕ_3}:	<p>Costa Concordia is a "Concordia"-class cruise ship for Costa Crociere. She was handed over to Costa Crociere on 29 May 2009. The total cost of the disaster, including victims' compensation, refloating, towing and scrapping costs, is estimated at approximately \$2 billion, more than three times the \$612 million construction cost of the ship.</p> <p>Comment: Miss the auxiliary information about the disaster.</p>

Table A.4: Qualitative examples on the reward model ablation experiment. We show the LM outputs when we use all the reward models and when one of the reward models is removed during training. We also mark the errors in each output.

A.4.4 Comparison with ChatGPT responses

We compare the responses generated by ChatGPT (one-shot)⁶ and our system in Table A.5. As discussed in § 6.4.6, We find that ChatGPT responses are relevant and factual, yet lack the auxiliary information to answer the ambiguous questions. This shows that it is challenging for ChatGPT to learn user-desired behaviors through prompting and in-context learning.

⁶Since the input for each example is very long, we cannot fit more than one in-context example into the model.

Question:	When did the Rams go to St Louis?
Passages:	<p>Article Title: History of the Los Angeles Rams</p> <p>The Los Angeles Rams are a professional American football team that play in the National Football League (NFL). The Rams franchise was founded in 1936 as the Cleveland Rams in the short-lived second American Football League before joining the NFL the next year. In 1946, the franchise moved to Los Angeles. The Rams franchise remained in the metro area until 1994, when they moved to St. Louis, and were known as the St. Louis Rams from 1995 to 2015. The Rams franchise returned to Los Angeles in 2016. This article chronicles the franchise’s history during their time in Los Angeles, from playing at the Los Angeles Memorial Coliseum between 1946 and 1979, to playing at Anaheim Stadium (now known as Angel Stadium of Anaheim) in Anaheim from 1980 to 1994, and its return to Southern California beginning with the season. On January 12, 1946, Dan Reeves was denied a request by the other National Football League (NFL) owners to move his team, the Cleveland Rams to Los Angeles and the then-103,000-seat Los Angeles Memorial Coliseum. Reeves threatened to end his relationship with the NFL and get out of the professional football business altogether unless the Rams transfer to Los Angeles was permitted. A settlement was reached and, as a result, Reeves was allowed to move his team to Los Angeles.</p> <p>Article Title: History of the St. Louis Rams</p> <p>The professional American football franchise now known as the Los Angeles Rams played in St. Louis, Missouri, as the St. Louis Rams from the 1995 through the 2015 seasons before relocating back to Los Angeles where the team had played from the 1946 season to the 1994 season. The Rams franchise relocated from Los Angeles to St. Louis in 1995, which had been without a National Football League (NFL) team since the Cardinals moved to Phoenix, Arizona in 1988. The Rams’ first home game in St. Louis was at Busch Memorial Stadium against the New Orleans Saints on September 10, 1995, before the Trans World Dome (later the Edward Jones Dome, and now known as The Dome at America’s Center) was completed for their November 12 game against the Carolina Panthers. Their last game played at the Edward Jones Dome in St. Louis was against the Tampa Bay Buccaneers on December 17, 2015, which they won, 31–23. The Rams’ last game as a St. Louis-based club was on January 3, 2016, against the San Francisco 49ers at Levi’s Stadium, where they lost in overtime 19–16.</p>
Comparison between our output and ChatGPT output	
ChatGPT:	The Rams moved to St. Louis in 1995 and were known as the St. Louis Rams from 1995 to 2015.
FINE-GRAINED RLHF	The Los Angeles Rams are a professional American football team that play in the National Football League (NFL). The Rams franchise was founded in 1936 as the Cleveland Rams in the short-lived second American Football League before joining the NFL the next year. In 1946, the franchise moved to Los Angeles. The Rams franchise remained in the metro area until 1994, when they moved to St. Louis, and were known as the St. Louis Rams from 1995 to 2015. The Rams franchise returned to Los Angeles in 2016.
Question:	Who sang the song I’m Coming Out?
Passages:	<p>Article Title: I’m Coming Out</p> <p>There’s also been a cover by Marcia Hines. It was featured as a cover for the video game "Karaoke Revolution" as well. In more recent times, the song was sampled in Keyshia Cole’s remake of "I’m Coming Out" (featuring Iggy Azalea) for the original soundtrack of the 2014 movie The Other Woman, starring Cameron Diaz, Kate Upton and Leslie Mann. The song is sampled in "Break Your Heart Right Back" from Ariana Grande’s sophomore album "My Everything" (2014). This song was also covered for the soundtrack to the 2016 movie "Trolls". In 1997, German comedian Hella von Sinnen sampled "I’m Coming Out" for her single "Mein Coming Out", using the term "coming out" in the context of coming out of the closet, or revealing one’s homosexuality. In 2003, American singer Amerie covered "I’m Coming Out" for the soundtrack to the film "Maid in Manhattan". Her version was released as a single in select European countries and Australia. The final single version to be released was the Loren Dawson remix. The original single version (which is similar to the original) can be found on both CD singles and the original motion picture soundtrack to "Maid in Manhattan". Ross’ version appears on the film’s soundtrack album as well.</p> <p>Article Title: I’m Coming Out</p> <p>"I’m Coming Out" is a song recorded by American singer Diana Ross. It was written and produced by Chic members Bernard Edwards and Nile Rodgers, and released in August 22, 1980 as the second single from Ross’ self-titled tenth album "Diana" (1980). In 1979, Ross commissioned Chic founders Nile Rodgers and Bernard Edwards to create material for a new album after taking her daughters to see the band in concert, following the success of Ross’s final Ashford & Simpson-composed and produced LP, "The Boss." "Rodgers got the idea for "I’m Coming Out" after noticing three different drag queens dressed as Diana Ross at a New York club called the GG Barnum Room. The lyrics hold an additional meaning to Ross, as she was leaving Motown Records and "coming out" from under Berry Gordy’s thumb. According to the BBC4 documentary "How to Make It in the Music Business", Ross loved the record until she put it in front of Frankie Crocker, who pointed out that 'I’m Coming Out' is what homosexuals use to announce their homosexuality; she ran back to the studio in tears and demanded to know why Rodgers wanted to ruin her career. The song was another hit from the album, peaking at number five on the U.S.</p>
Comparison between our output and ChatGPT output	
ChatGPT:	The song "I’m Coming Out" was recorded by American singer Diana Ross. It was written and produced by Chic members Bernard Edwards and Nile Rodgers, and released in August 22, 1980 as the second single from Ross’ self-titled tenth album "Diana" (1980).
FINE-GRAINED RLHF	"I’m Coming Out" is a song recorded by American singer Diana Ross. It was written and produced by Chic members Bernard Edwards and Nile Rodgers, and released in August 22, 1980 as the second single from Ross’ self-titled tenth album Diana. In 2003, American singer Amerie covered "I’m Coming Out" for the soundtrack to the film Maid in Manhattan. Her version was released as a single in select European countries and Australia. The final single version to be released was the Loren Dawson remix.

Table A.5: Comparison between ChatGPT outputs and FINE-GRAINED RLHF outputs.