Reward-free Policy Imitation Learning for Conversational Search

Zhenduo Wang
zhenduow@cs.utah.edu
University of Utah

Zhichao Xu
zhichao.xu@utah.edu
University of Utah

Qingyao Ai
aiqy@tsinghua.edu.cn
Tsinghua University

Abstract

Existing conversational search studies mainly focused on asking better clarifying questions and/or improving search result quality. These works aim at retrieving better responses according to the search context, and their performances are evaluated on either single-turn tasks or multi-turn tasks under naive conversation policy settings. This leaves some questions about their applicability in real-world multi-turn conversations where realistically, each and every action needs to be made by the system itself, and search session efficiency is often an important concern of conversational search systems. While some recent works have identified the need for improving search efficiency in conversational search, they mostly require extensive data annotations and use hand-crafted rewards or heuristics to train systems that can achieve reasonable performance in a restricted number of turns, which has limited generalizability in practice.

In this paper, we propose a reward-free conversation policy imitation learning framework, which can train a conversation policy without annotated conversation data or manually designed rewards. The trained conversation policy can be used to guide the conversational retrieval models to balance conversational search quality and efficiency. To evaluate the proposed conversational search system, we propose a new multi-turn-multi-response conversational evaluation metric named Expected Conversational Reciprocal Rank (ECRR). ECRR is designed to evaluate entire multi-turn conversational search sessions towards comprehensively evaluating both search result quality and search efficiency.

CCS Concepts

• **Information systems → Users and interactive retrieval**.

Keywords

conversational search, imitation learning, conversational search evaluation

1 Introduction

Conversational AI is one of the long-standing research goals in both natural language processing (NLP) and information retrieval (IR) communities as well as industry. Along with the development of conversational agents such as Microsoft Cortana, conversational search and other related research topics such as conversational question answering, task-oriented dialog systems, social chat, and not the least conversational recommendation have drawn many interests in the past few years [8, 29, 73, 82]. Among these topics, conversational search is a retrieval-based approach for conversational information seeking. It retrieves responses given the initial query and current conversation context [78]. Most existing conversational search studies and datasets [30, 43, 50, 69, 70, 81] cast conversational search as a next-response ranking (NPR) task where all types of responses (e.g., returning results, and asking clarifying questions) are indiscriminately sampled and ranked together as response candidates.

Recent studies about conversational search conceptualization [7, 55], utterance intent identification [53, 54], and asking clarifying questions [4, 12, 38] have suggested that modeling conversational search as naive sequence of responses as in NPR maybe not be the most effective way. Rather, it would be better to categorize user and system utterances separately using fine-grained conversation action spaces. For example, user can perform actions such as inquiry, refinement, and closing, while system can perform actions such as revealing and clarification. This naturally requires the system to have the ability to decide which of these actions to perform (conversational search policies), which leads to a system architecture that is very similar to a task-oriented dialog system [25], and MACAW [74] is a virtual example of such system architecture. However, within the scope of conversational search, only a few works have been proposed in this direction, or at least have addressed the need of making conversation action decisions, such as [2, 3, 80]. Wang and Ai [62, 63] studied the risk of conversational search and proposed to solve conversational search policy as a risk-controlling task. However, the risk of asking clarifying questions is not the only factor of conversational search policy. Other factors such as how to maximize search efficiency and improve user engagement during conversation should also be considered. Hence, their conversational search policy could be risk-biased and overlook the opportunities to improve search result quality.

There are more studies about dialog policy learning in the task-oriented dialog systems community. Many existing works about policy learning employ reinforcement learning (RL) algorithms and train a dialog policy model through interactions with user simulator models [19, 20, 49, 58]. However, designing reinforcement learning rewards is a big challenge in these works since it requires a large amount of knowledge about the task and dataset. Poorly designed rewards can often lead to undesired system behaviors [59]. In order to achieve good performances, most of these works manually tune their rewards, which consequently limits their generalization ability.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

In this paper, we propose a novel reward-free conversational search policy imitation learning (IL) framework. Our IL framework trains a simple conversation policy model, which decides between asking the retrieved clarifying question and returning the retrieved results to the user at each conversation stage, given the current conversation context and retrieval results. Inspired by recent advances in Generative Adversarial Imitation Learning (GAIL) [31], to train the conversation policy model, we compute and identify an expert trajectory (a multi-turn-multi-response tuple) from all possible conversation trajectories using an automatic evaluation metric, and we use the expert trajectories as the training samples without any data annotations or manually-crafted rewards. Our IL framework also allows us to train policies according to different conversational search evaluation metrics that represent different user assumptions. Hence, our policy learning framework could potentially generalize to other future conversational search tasks, datasets, and user assumptions.

To better evaluate the multi-turn-multi-response conversational search sessions, we also propose a new automatic conversational evaluation metric called Expected Conversational Reciprocal Rank (ECRR). Our metric can evaluate the entire response trajectory of the search agent in the multi-turn conversational search session. This is fundamentally different from naive metrics which evaluate one single-turn retrieved response quality at a time. Our metric is also an attempt toward a comprehensive evaluation metric for conversational search, which evaluates both the search result quality and search session efficiency.

We consider our work has three contributions:

- We propose a new reward-free conversational search policy imitation learning framework, which can learn a conversation policy to jointly optimize search effectiveness and efficiency without data annotations or manually designed rewards.
- To comprehensively evaluate search effectiveness and efficiency, we propose a new multi-turn-multi-response conversational search evaluation metric named Expected Conversation Reciprocal Rank, which can better approximate system performances in real-world scenarios.
- To the best of our knowledge, we are also the first to use multi-turn metrics directly as an indirect learning goal of conversational policy learning.

The rest of the paper is organized as follows: we first review the related work (§2); then introduce the proposed IL-based framework (§3) and propose to evaluate conversational search task with the proposed ECRR metric (§4). We cover the detailed experimental setup (§5), report and analyze the experimental results (§6). Finally, we wrap this work up and point out future directions (§7). To facilitate reproducibility of the IR community, our code will be made public once this work is accepted.

2 Related Works

Conversational Search Studies on conversational search can be traced back to the very beginning of research on interactive information retrieval [11, 17, 47]. The framework proposed by Radlinski and Craswell [55], along with other conceptualization works including [7, 18] is some examples of the recent surge of conversational search system studies. These works have taken various approaches and covered multiple aspects of conversational search. For example, Yu et al. [71, 72] study effective query rewriting and learning contextualized query representation from an ad hoc teacher model. Zhang et al. [80] study conversational preference elicitation. Vakulenko et al. [61] study knowledge-based conversational search. Some works [4, 16, 70, 76] provide datasets with different focuses for these studies. Other works such as [6, 22, 23, 29] provide seminars and tutorials from different standpoints.

Among these topics, a popular line of work (e.g. [4, 56, 77]) about asking clarifying questions is highly related to this paper. Asking clarifying questions in conversational search can be traced back to the TREC 2004 HARD track [5], where asking clarifying questions is a system option to get additional information. Recently, more works have shown that asking clarifying questions can benefit conversational search systems. A study in 2018 [38] showed that users enjoyed interacting with conversational systems. Following this work, several other works [4, 51, 75] demonstrate that asking clarifying questions is a convenient alternative when the query information is insufficient for retrieving good results. We may need to reconsider the question today that how long we can be optimistic about users' tolerance and patience for the mistakes of our conversational systems after their freshness for conversational AI, as interacting with conversational AI will soon if not has already become a norm of human life. However, only a few works [2, 62, 63, 68] have addressed the problem of deciding whether to ask clarifying questions in conversational search systems. Our work extends these works and generalizes the problem of identifying the need of asking clarifying questions to conversational search policy modeling.

Conversational Search Evaluation The evaluation of conversational search remains open and diverse because of (1) conversational intelligent system being a common research interest for both natural language processing and information retrieval community, (2) the term-sharing between conversational search and other related areas like dialog system and conversational QA, (3) the variety of task configurations, solutions, and evaluations (e.g. slot filing/response retrieval/generation, single/multiple responses per turn, single/multi-turn evaluation), and more. Due to the cost and complexity of involving human users in the loop, apart from a few works which study the online evaluation framework and interface [7, 36, 37], most works simulate users' behaviors and evaluate the conversation sessions with offline methods for its simplicity and reproducibility. Metrics evaluated on single-turn single-response can be categorized as word-overlap-based (BLEU [48], METEOR [9]), embedding-based (BERT-Score [79]), learning-based (BERT-RUBER [26]), or F1 score for slot filing tasks. Metrics evaluated on single-turn multi-response are mainly ranking-based metrics (nDCG [34], RBP [45], ERR [14]).

The evaluation of conversational search still faces many challenges [51]. Metrics evaluated on multi-turn sessions are usually done by combining single-turn metrics with or without session-based weighting (SWF [42], sDCG [34]). As a matter of fact, the adaption of single-turn evaluation metrics for multi-turn evaluation is actually more complicated than it shows in the above methods. The reason is that a conversational search system which, by its definition, will interact with the user and generate indefinite search sessions, which means that different system responses can lead to completely different conversation trajectories. Thus combining single-turn evaluation metrics through weighting seems questionable, as it overlooks the effect of each turn on future turns due to the turn-independence assumption. Recently, there have been some works about multi-turn evaluation. sRBP [40] evaluates session rank based on user model. Wang and Ai [62, 63] evaluate conversational search sessions by simulating with various types of users. ECS [41] suggests to evaluate entire conversation session by identifying sub-topic. We follow the same setup with these works and use a cascade user model for multi-turn evaluation.

While most of the above metrics are focused on evaluating result quality or search effectiveness, other works also mention measuring search efficiency. Optimizing search efficiency (or search effort) means promoting more efficient conversational systems which can achieve the same goal with less interaction with the user, and are usually measured by the number of conversation turns. Early works like [28, 35] highlight efficiency as another major measure beside effectiveness in interactive QA evaluation. Among more recent works, RBP [45] and ERR [14] are two single-turn multi-response evaluation metrics with efficiency measuring aspect. They are both based on an explicit user behavior simulator that models the user's patience during the search in exhausting the search result list.

Task-oriented Dialog System With the ability to interact with the user, conversational search systems are closely related to dialog systems, as they both aim to help the user through multi-turn conversations. However, a task-oriented dialog system usually knows the exact task it needs to solve, such as flight booking or weather query, while conversational search tasks are more open. One way to connect the two tasks and think about their similarities is to regard conversational search as a goal-oriented dialogue, where its goal is information seeking [25]. Anand et al. also [6] type conversational search system as *an information-seeking dialogue system with information retrieval capabilities*. Works like these have shown that conversational search and task-oriented dialog systems are highly similar in many aspects. Recently, a system-ask-user-respond scheme that is common in task-oriented dialog systems is also seen in conversational search and recommendation system studies like [4, 80]. While task-oriented dialog systems usually respond to the user by generating natural language responses, conversational search systems focus on retrieving relevant information from massive web sources to the user. Besides this difference, a task-oriented dialog system can often involve solving multiple sub-questionnaires of the task, while the conversational search problem we focus on in this paper is to clarify and answer the user's initial information request through multi-turn conversation.

Because of the connections between task-oriented dialog systems and conversational search systems, these systems share many structural similarities. Previous research on task-oriented dialogue systems in NLP can be roughly categorized into two groups [24, 82]: (1) pipeline/modular system, and (2) end-to-end system. Pipeline systems regard the process of a task-oriented dialogue system as an iterative decision-making process with four major modules: natural language understanding (NLU), dialogue state tracking (DST), dialogue policy (POL), and natural language generation (NLG). During the iterative decision-making process, the system reads the dialogue (NLU) and processes the information to have an understanding of the current dialogue state (DST), then decides its next action concerning the understanding (POL), and finally generates natural language output implementing the action (NLG). Some works try to combine some of the modules such as combining NLU with DST [46, 65], or policy with NLG [15, 83].

End-to-end systems such as [13, 39, 64] model task-oriented dialogue as a response generation problem through language modeling, with the entire dialogue history modeled as a long text sequence, essentially serializing the pipeline.

Imitation Learning Imitation learning (IL) aims to train a system to mimic human behavior by learning from human demonstrations [33]. IL can be implemented as supervised learning, where the agent learns to behave like the expert on all training samples. This is known as the behavior cloning method [52]. However, simply copying the behavior on the training set has been shown to be inefficient and not generalizable. IL is more often cast as an inverse reinforcement learning (IRL) problem [1, 66, 67, 84], where the system tries to infer the reasoning/logic behind these expert demonstrations, then learn to apply them to unseen scenarios to generalize. In general reinforcement learning (RL) algorithms, it is usually required to have explicit or manually designed rewards signals for training. However, in real-world tasks such as automatic driving and robotics, such signal rarely exists and defining rewards is usually difficult or even impossible. IRL is a learning paradigm to learn a policy with no available rewards by inferring the rewards from expert demonstrations for the RL algorithm to use.

The IRL algorithms and their deep neural network versions iteratively perform the IRL-RL learning steps from expert demonstrations and are generally considered to be costly in terms of time. With the advance of studies about generative adversarial nets (GAN) [27], Finn et al. [21] suggest the equivalencies between GAN, IRL, and energy-based models, and Ho et al. [31] propose that the IRL-RL learning iteration can be simplified by solving the dual problem of occupancy measure matching. They subsequently propose a generative adversarial imitation learning (GAIL) algorithm, which could reduce the cost of imitation learning by a large margin. Till today, GAIL has been studied and extended by many works [10, 57, 60].

The applications of imitation learning methods can mostly be found in robotics and autonomous driving. To the best of our knowledge, our work is among the very first efforts to apply imitation learning methods to conversational search system studies.

3 Proposed System

We start this section with an overview of our conversational search system (§3.1); then we walk through the system from retrieval models (§3.2) to the policy model (§3.3). Finally we introduce the inference of the proposed model (§3.4).

3.1 Conversational Search System with Policy

Our conversational search system models a conversational search agent which aims to retrieve relevant information for the user's information need by interacting with the user through multi-turn conversation. Our system consists of two modules:

The first module is a collection of retrieval models, each modeling a specific action that the agent may take, e.g. retrieving results, or asking clarifying questions. We assume that most conversational search systems such as [2, 68] have the same or similar structure as the first module. However, most of them do not have a model to decide which action to make, which is our second module.

The second module is a conversational search policy model, which decides the next agent action given the user's search query, current conversation context, and the retrieved candidates of each action from the first module. In our work, we only consider two main agent actions, returning the retrieved result to the user or asking a clarifying question to the user. Therefore, our policy model only needs to make decisions between the two actions, and there will be only two retrieval models, namely the result retriever and the clarifying question retriever. However, given enough community attention and studies on other possible agent actions and related datasets, our agent should easily be extended and adapted. Our conversational search agent is also an implementation of recent conversational search conceptualization studies, which suggest categorizing agent actions and modeling them individually as discussed in Section 1.

It has been proposed and shown in [62, 63, 74] that integrating the retriever outputs in the policy model can improve the overall agent performance and control the risk in clarifying questions compared to the popular baseline of first predicting agent action and then running the corresponding retrieval model. Following their works, our system also runs the retrieval models before the policy model. Hence, we will first introduce the retrieval models (§3.2) and then the policy model (§3.3).

3.2 Retrieval Models

We have two retrieval models for modeling the search result retrieval and clarifying question retrieval tasks, respectively. The goal of the retrieval models is to retrieve the best result or clarifying question-based on the search query and conversation context. The result and clarifying question retriever use the same poly-encoder [32] structure but do not share parameters and are trained separately.

Poly-encoder Assume the conversation context between user and agent can be represented as $\{U_0, A_0, .., U_N, A_N\}$, where Us are user utterances, and As are agent utterances. The response candidates can be represented as $P = \{p_1, .., p_K\}$. In poly-encoder, the appended search context $q =$ `[S]`U_0`[SEP]`A_0`[SEP]`U_1... is first encoded into vectors $(h_{q_1}, .., h_{q_{2N}})$ with pretrained transformer, where `[S]` represents the start of sentence token, and `[SEP]` represents the sentence segmentation token. Then it attends to m learnable codes $(C^1, .., C^m)$ to generate m attended context vectors $(q^1, .., q^m)$:

$$q^i = \sum_j^{2N} w_j^i h_{q_j} \tag{1}$$

where $(w_1^i, .., w_{2N}^i) = \text{softmax}(C^i \cdot h_{q_1}, .., C^i \cdot h_{q_{2N}})$.

The poly-encoder then encodes each candidate p_k into a vector E_{p_k} using pretrained transformer T and dimension reduction function $red(\cdot)$, which can aggregate a sequence of vectors into one vector:

$$E_{p_k} = red(T(p_k)) \tag{2}$$

Finally, the m context vectors $(q^1, .., q^m)$ attend to the candidate vector E_{p_k} to compute candidate-attended context vector E_q:

$$E_q = \sum_i^m w_i q^i \tag{3}$$

where $(w_1, ..., w_m) = \text{softmax}(E_{p_k} \cdot q^1, .., E_{p_k} \cdot q^m)$. Then the ranking score of candidate p_k is computed as the dot product $s_k = E_{p_k} \cdot E_q$. The output of the retriever is the rank of all the candidates and their ranking scores.

The result retriever and clarifying question retriever are trained separately. Both of them are first initialized using pretrained checkpoints, and then fine-tuned on batches of `(search context, result)` or `(search context, clarifying question)` pairs to minimize the cross-entropy loss between the softmax result of ranking score vector and the true relevance label vector.

$$L_{\text{poly}} = \sum^B \text{CE}(\text{softmax}(s_1, .., s_K), I(p_1, .., p_K)) \tag{4}$$

After the retrieval models are trained, their parameters will be fixed during the training of policy models.

3.3 Policy Model and Imitation Learning

Markov Decision Process We consider the task of choosing which retrieval result to return in each stage of conversational search as a Markov Decision Process (MDP) problem. The conversational search MDP is a tuple $\{\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}\}$ where:

- $\mathcal{S}$ is a set of conversational states
- $\mathcal{A}$ is a set of system actions
- $\mathcal{T}(s, a)$ is a transition probability distribution that an action a in state s will lead to s'
- $\mathcal{R}(s, a)$ is the immediate reward of taking action a in state s

To be more specific, the state $\mathcal{S}$ comprises the initial query, utterance history context, and retrieved clarifying questions and results. The system action set $\mathcal{A} = \{\text{return results, ask question}\}$. The transition probability $\mathcal{T}$ is modeled by a user cascade model, which will be explained in section 5

A trajectory τ in MDP is a series of system actions together with their resulting states. τ can be efficiently represented as $\tau = \{(s_1, a_1), .., (s_t, a_t)\}$. In conversational search, the system only has two actions. We assume that the action of returning results will end the conversation, and the action of asking a question will continue the conversation. Therefore, any conversational search trajectory will be an arbitrary number of asking-question actions followed by one returning-result action.

Policy Model The job of our conversational search policy model is to make decisions of which agent action to take, at any state S. In our setting, the agent only has two actions, either retrieving result for user's query or asking a clarifying question. Hence, the output of our policy model is a probability distribution over the two actions. To represent the input state S, we first use pretrained transformer to encode the initial query, iq, the current conversation utterances $q =$ `[CLS]`U_0`[SEP]`A_0`[SEP]`U_1..., the retrieved result candidates $(re_1, .., re_K)$, and the retrieved clarifying question candidates $(cq_1, .., cq_K)$. Then, we concatenate all these encodings together with the retrieved result and clarifying question scores as

Algorithm 1 LSGAIL for conversational search policy learning

```
**Input** Expert trajectories $\tau_E \sim \pi_E$, initialized policy $G_{\theta_0}$, initialized discriminator $D_{\sigma_0}$
**for** $i = 0, 1, 2, ...$ **do**
    Sample trajectory $\tau_i \sim \pi_{\theta_i}$
    Update $\sigma_i$ by minimizing target:
        $\sum_{\tau_i} D_{\sigma_i}(s,a)^2 + \sum_{\tau_E}(D_{\sigma_i}(s,a) - 1)^2$
    Update $\theta_i$ by maximizing KL-constrained target:
        $\mathrm{E}_{\tau_i}[\log(G_{\theta_i}(a|s) * Q(s,a))] - \lambda H(\pi_{\theta_i})$,
        where $Q(s,a) = E_{\tau_i}[\log(D_{\sigma_i}(s,a))|s_0 = s, a_0 = a]$
**End for**
```

the state representation $S = (iq, q, re_1, .., re_K, cq_1, .., cq_K, s_{re}^{1:K}, s_{cq}^{1:K})$. The policy model G is a 2-layer feed-forward neural net:

$$G(S) = \text{softmax}(W_{G2} \cdot \phi(W_{G1} \cdot S + b_{G1}) + b_{G2}) \tag{5}$$

where $W_{G1}, W_{G2}, b_{G1}, b_{G2}$ are learnable parameters.

Learning Policy models are usually trained by reinforcement learning methods such as policy gradient or Q-learning, which require manually designing and tuning the rewards function of each action. For example, Wang and Ai [63] empirically determine the reward of asking clarifying questions and returning answers with the intention to control the risks of agent actions. However, this could lead to a rather conservative conversational search policy, which will further weaken the possibility of improving result quality by asking more clarifying questions.

RL algorithms are notoriously hard to train due to their requirement of manually designing and tuning rewards functions. To avoid this effort, we propose to use imitation learning (IL), which is to learn the policy directly from the expert demonstrations of the task without the help of additional training signals. The imitation learning algorithm we use is Generative Adversarial Imitation Learning (GAIL), which alternatively updates and optimizes a (state, action) evaluator as a discriminator D and the policy model as a generator G. We use GAIL instead of other IL algorithms as it can best approximate the expert policy by minimizing a real metric defined on policy space and is more efficient compared to other IL algorithms [31]. The algorithm suited for our task is briefed in Algorithm 1. Compared to the original GAIL, we change the discriminator loss function to least square objective [44] for better performance. Therefore, we denote our framework as Least Square-GAIL or LSGAIL.

GAIL algorithm requires expert demonstration τ_E as input, which is a trajectory of `(state, action)` pairs that can provide the best conversational search quality. τ_E can be represented as $\tau_E = \{(s_0, a_0), .., (s_T, a_T)\}$. However, most conversational search datasets only have the raw conversations which can be represented as $\{U_0, A_0, .., U_N, A_N\}$. To get τ_E, we evaluate all possible conversational search trajectories in terms of resulting conversational search quality and select the best as the expert demonstration. The metric we use to evaluate the conversational search quality will be discussed in §5. The sampled trajectory τ_i is the trajectory decided by the policy model parameters in the current iteration.

The input of the `(state, action)` discriminator D in the GAIL algorithm is the concatenated vector S_a of the initial query iq, current utterance history q, the candidate text of the action, and the candidate ranking scores s of the action. For example if the action is to return the result, then $S_a = (iq, q, re_1, ...re_K, s_{re}^{1:K})$. If the action is to ask the clarifying question, then $S_a = (iq, q, cq_1, ...cq_K, s_{cq}^{1:K})$. The output of D is a scalar indicating the probability of the `(state, action)` pair being from the expert trajectory. D is a 2-layer feed-forward neural net:

$$D(S_a) = \text{Sigmoid}(W_{D2} \cdot \phi(W_{D1} \cdot S_a + b_{D1}) + b_{D2}) \tag{6}$$

where $W_{D1}, W_{D2}, b_{D1}, b_{D2}$ are learnable parameters.

Our policy module is theoretically superior to previous RL-based conversational search models [62, 63] as the training of discriminator D and policy G is reward-free, thus avoiding the burdensome reward designing and tuning steps. The idea behind GAIL is to train D with expert demonstrations and demonstrations generated by G, and use $\log(D(s,a))$ as the estimated reward during the update of G.

3.4 Inference

During inference, an initial user query is first fed into the conversational search system; the system then enters the interaction loop with the user, where it will decide and make an action between asking clarifying questions and returning the search result to the user. The result and question retrieval models are run once to generate the input for the policy model, and then the policy model will make the final decision. The interaction loop will continue if the user responds to the clarifying question, and will stop if the system returns the search result or its retrieved clarifying question list does not contain any useful questions. Finally, the output of our system is a simulated conversational search trajectory which is a series of clarifying question and search result retrieval results and can be represented as $\{\pi_{cq}^{1:T-1}, \pi_{res}^T\}$, where $\pi_{cq}^{1:T-1}$ represents the retrieved clarifying question ranklists for the first $(T-1)$ clarifying turns, and π_{res}^T is the retrieved result ranklist for the last turn. We will evaluate our system using the entire trajectory it produces.

4 Evaluation Metrics

4.1 Expected Conversational Reciprocal Rank

In §3, we briefly mentioned an evaluation metric we use to select the expert demonstration from all possible conversation trajectories. We will explain this metric which we use for both selecting the expert demonstration and evaluating the performance of our system and the baseline models during inference.

The output of our conversational search system is a conversational search trajectory which can be represented as $\{\pi_{cq}^{1:T-1}, \pi_{res}^T\}$, where $\pi_{cq}^{1:T-1}$ represents the retrieved clarifying question ranklists for the first $(T-1)$ clarifying turns, and π_{res}^T is the retrieved result ranklist for the last turn. With the above notations, we define the evaluation metric Mean Expected Conversational Reciprocal Rank based on Cascade model (Mean ECRR) as:

$$\text{Mean ECRR} = \frac{1}{M}\sum_{i=1}^{M} \text{ECRR}_{\mathcal{U}}(\pi_{cq}^{1:T-1}, \pi_{res}^T)$$

where ECRR is the Expected Conversational Reciprocal Rank for single conversation, and $\mathcal{U}$ is a cascade model to simulate user behavior in response to a retrieved clarifying question list. This is very similar to previous works such as [14], which also uses a user cascade model in their metric. To better understand cascade model and ECRR, we recommend imagining the term *user* not as a single user but the entire user base instead. The cascade model assumes that (1) user will examine the clarifying question list by its rank order, and have a probability α to examine each clarifying question; (2) when seeing a relevant clarifying question, user will answer the question and wait for next response from the conversational system; (3) when seeing an irrelevant answer or clarifying question, user will continue to examine the next response with probability α or leave the conversation with probability $(1 - \alpha)$. The parameter α reflects the percentage of users we assume to be tolerant and patient in the user base.

With the cascade model assumption, during all but the last conversation turn $t \leq (T - 1)$, when a retrieved clarifying question ranklist π_{cq}^{t} is returned to user, we can compute the following probability

$$p^{t} = \alpha^{r(\pi_{cq}^{t}=1)} \tag{7}$$

This is the probability that user will examine the true relevant clarifying question using cascade model on π_{cq}^{t}, where $r(\pi_{cq}^{t} = 1)$ denotes the rank of relevant clarifying question. Then, user will either examine the clarifying question and continue the conversation with this probability p^{t}, which will leave the quality of the conversation to be further determined by future turns, or leave the conversation with probability $(1 - p^{t})$ without seeing a result, which is a disappointing scenario for both the user and our system. The above process can be described using the following equation:

$$\text{ECRR}^{t} = p^{t} \cdot \text{ECRR}^{t+1} + (1 - p^{t}) \cdot 0 \tag{8}$$

In the last turn, when a retrieved result ranklist π_{res}^{T} is returned to user, we can finally evaluate the retrieved result ranklist quality using reciprocal rank of the true relevant result:

$$\text{ECRR}^{T} = \text{RR}(\pi_{res}^{T}) \tag{9}$$

From equation (1) and (2), we can derive that the ECRR of the entire multi-turn conversation is computed as:

$$\text{ECRR}_{cas}(\pi_{cq}^{1:T-1}, \pi_{res}^{T}) = \prod_{t=0}^{T-1} \alpha^{r(\pi_{cq}^{t}=1)} \cdot \text{RR}(\pi_{res}^{T}) \tag{10}$$

As mentioned earlier, the Ubuntu Dialog Corpus dataset only contains raw conversations. To train the imitation learning framework, we need to generate and compute expert trajectories from the raw data using ECRR. This can be done in the following steps (using one conversation as an example):

1. In each turn of the conversation, we run both of the retrieval models to generate a result rank list π_{res} and clarifying question rank list π_{cq}.
2. Assume the conversation has T turns, then there are T possible trajectories $\{\{\pi_{cq}^{1:t-1}, \pi_{res}^{t}\}^{t=1:T}\}$ in total, each ends at one of the T turns. We use formula (9) to compute the ECRR of all these trajectories, and then select the trajectory with the highest ECRR as the expert trajectory.

   Notice that different α can lead to different expert trajectories. For example, if we have a 2-turn conversation, and the retrieved results reciprocal rank for the 1st and 2nd turn are 0.33, 1, and the retrieved question rank for the 1st turn is 3. Using ECRR with $\alpha = 0.5$, the expert trajectory is to not ask the clarifying question, because asking the question has ECRR $= 0.5^{3} \times 1 = 0.125 < 0.33$. Using ECRR with $\alpha = 0.7$, the expert trajectory is to ask the clarifying question, since $0.7^{3} \times 1 = 0.343 > 0.33$.

ECRR can be seen as the average user satisfaction given the conversational search system response trajectory. This metric evaluates the full trajectory without assuming any specific user type. The entire user population is modeled by the parameterized cascade model. Some users may go through the full trajectory and receive a search result and other may not. Because of this, this metric is more realistic and applicable compared to the metrics used in [63].

According to the definition of ECRR (Equation 10), the final score is determined by the number and quality of clarifying questions (cumulative product term) and the final retrieval result quality (reciprocal rank term). A policy trained to optimize ECRR is encouraged to get the best retrieved result (high effectiveness) while minimizing the number of clarifying questions asked to user (high efficiency). Thus ECRR accounts for both search effectiveness and efficiency. The parameter α in the cascade model of ECRR controls the trade-off between effectiveness and efficiency.

As discussed in section 2, most existing works about conversational search system use single-turn metrics or stack single-turn metrics under naive conversation policy, our work is among the few works which evaluate conversational search with multi-turn metrics. Among these few works, the majority of them uses multi-turn evaluation metrics only for evaluation, while we propose to also use the evaluation metric during conversational search policy training.

4.2 Recall and MRR

Besides the ECRR metric, we also include recall and MRR as other evaluation metrics, which are more commonly used when evaluating ranked lists. However, it is important to notice that we use them on the entire conversational search system trajectory $\{\pi_{cq}^{1:T-1}, \pi_{res}^{T}\}$, instead of any single turn ranking. In this case, it is also worth mentioning that MRR is a special case of ECRR when the cascade user model has **binary** α. Specifically, $\alpha = 1$ when the top question is relevant, and $\alpha = 0$ otherwise. which means that users will always leave on seeing irrelevant clarifying questions.

5 Experiments and Evaluations

5.1 Experiment Design

The goal of our experiments is to show that the proposed reward-free IL framework can (1) work reasonably well without reward tuning, and (2) generalize to different evaluation metrics or user assumptions. To show (1), we compare our framework with a risk-aware conversational search policy model Risk-aware Conversational Search agent with Q-learning (RCSQ) [63], which needs manually tuning the reward. We will denote this model as RCSQ for the rest of our paper. With the retrieval models fixed, we run our framework and the RCSQ model with different rewards and compare their performances using various evaluation metrics. To

Table 1: Original Ubuntu Dialog and our dataset statistics

item	Ubuntu Dialog	Ours
#conversations	930,000	10,000
Max. turns	-	10
Min. turns	3	4
Avg. turns	7.71	4.77
Avg. #words per utterance	10.34	20.85

show (2), we train our conversational search policy regarding different evaluation metrics and compare its performance with the RCSQ model performance on these different evaluation metrics.

5.2 Dataset

We use the Ubuntu Dialog Corpus (UDC) dataset [43] in our experiments. The UDC dataset consists of question answering conversations about the Ubuntu system. We adopt the same processed and filtered dataset and simulation experiments as [62, 63]. The processing and filtering of the UDC dataset ensure that (1) all the conversations involve only two participants, i.e., the user and Microsoft agents, and in turns, (2) all the conversations have at least two turns, which means that there will be at least one clarifying question in each conversation. We use the filtered and randomly sampled 10000 conversations from [63] as our dataset. Further details of the dataset can be found in Table 1.

5.3 Baselines

Following our experiment design, we include two groups of baseline models in our experiments. The first group comprises 3 naive conversational search policies and 1 simple policy learning model:

(1) **Q0A**, a baseline that always returns results without asking a clarifying question. Hence, it always uses the initial query as the only information for answer retrieval.
(2) **Q1A**, a baseline that always returns results after asking exactly one clarifying question. It will always ask a clarifying question in the 1st turn, and then return results.
(3) **Q2A**, a baseline similar to Q1A, but it will always return results in the 3rd turn after asking clarifying questions in the first two turns.
(4) **CtxPred**, a simple conversational search policy learning model using behavior cloning [52]. This is similar to the clarifying-question-need classification models in works such as [2, 68].

The second group of baselines comprises a few untuned variations of the RCSQ model in [63]. We use RCSQ as a representative for reinforcement learning policy models in general. The meaning of having untuned versions of RCSQ is for the comparison between the proposed reward-free IL framework and reward-tuning methods such as RCSQ.

Finally, we include an oracle policy that can foresee all possible conversation trajectories and can always take the best action by backtracking. It is worth mentioning that a conversational search policy does not improve the retrieval models it works with. It only decides which retrieval model output (results or questions) to return to the user. Hence, given fixed retrieval models, there will be an upper bound for any conversational search policy. This oracle policy is the upper bound and is only meant to be used for reference.

Table 2: RCSQ model reward table

	Relevant	Irrelevant
Search results	Result Reciprocal Rank	
Clarifying question	$r_{cq} = 0.11$	$p_{cq} = -0.89$

5.4 Technical Details

We use the poly-encoder implementation from ParlAI[1]. We download their pretrained checkpoints and finetune them on the UDC dataset. We implement our proposed IL framework from scratch based on Pytorch. We use the RCSQ model implementation from their GitHub repository. Our main experiments are run on a single-core GeForce RTX 2080 Ti with 11GB of memory. The pre-training of the poly-encoder retrievers is done on 4 of the above cores with batch size = 8.

In our experiments, we use the split dataset of train/val/test sets using 8:1:1 ratio in [63]. The number of negative samples $k = 99$. The original RCSQ model uses $r = 0.11$ as their tuned reward. We test their model with untuned rewards $r = -0.1, 0.3, 0.5, 0.7, 0.9$. We train our model with a learning rate $lr = 10^{-4}$, and entropy weight $\lambda = 10^{-2}$. In each epoch of GAIL training, we will train the discriminator 5 times and the generator 1 time.

6 Results and Analysis

6.1 Experiment Results

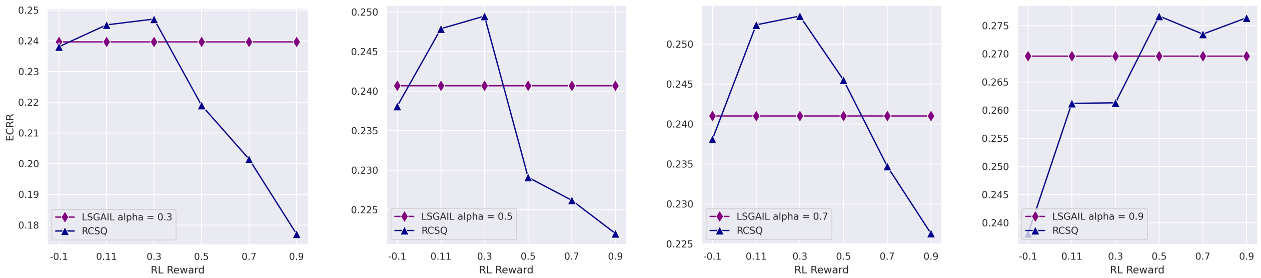
Our experiment results are shown in Table 3. The rows are different conversational search policies. The 1st to 4th rows are the naive baseline models. Q0A, Q1A, and Q2A are determined policies, which ask exactly 0, 1, and 2 questions respectively before returning results to the user. CtxPred is a behavior cloning baseline that is trained using expert trajectories. These 4 baseline models in the first block represent the performances when using the state-of-the-art dense retrieval models directly without a conversational search policy. The second block is the untuned risk-control policies [63] denoted as RCSQ. For example, the row of RCSQ $r = 0.11$ is RCSQ with a reward table shown in Table 2. The third block is our proposed IL framework trained regarding different training targets, denoted as LSGAIL. For example, the row of LSGAIL $\alpha = 0.3$ is LSGAIL trained under ECRR with cascade $\alpha = 0.3$, where α is the frequency that the user will remain in the conversation after seeing an irrelevant clarifying question. The last row is the oracle policy, which is the performance upper bound of any conversational search policy, given the fixed retrieval models. This performance upper bound is only meant to be used for reference, since it is almost unachievable.

The columns are different evaluation metrics. R@1/100 means recall@1 from 100 candidates, and MRR is the mean reciprocal rank. Notice that they are both computed on the entire conversational search trajectory. Followed by them are 4 different ECRRs with $\alpha = 0.3, 0.5, 0.7, 0.9$. From the table, we can see that each LSGAIL variation performs the best on ECRR with the same α (bolded in the table). This shows that different LSGAIL variations are trained

[1] https://github.com/facebookresearch/ParlAI/tree/master/projects/polyencoder

Table 3: Comparison of all models and baselines on sampled UDC dataset using poly-encoder as re-ranker. GAIL model is trained using ECRR cascade user model $\alpha = 0.3, 0.5, 0.7, 0.9$. ρ is the user patience for total clarifying questions, τ is the user tolerance for irrelevant clarifying questions. Numbers in bold mean the result is the best among all the variation. † and ‡ means $p < 0.1$ and $p < 0.05$ statistical significance over the all other models. ⋆ denotes the model variation used by [63].

Policies	R@1/100 (binary α)	MRR (binary α)	ECRR $\alpha = 0.3$	ECRR $\alpha = 0.5$	ECRR $\alpha = 0.7$	ECRR $\alpha = 0.9$
Q0A	**0.1580**	**0.2381**	**0.2381**	**0.2381**	**0.2381**	0.2381
Q1A	0.1200	0.1630	0.1799	0.1938	0.2214	**0.2693**
Q2A	0.0230	0.0284	0.0359	0.0387	0.0530	0.0761
CtxPred	0.1165	0.1581	0.1531	0.1841	0.1951	0.2182
RCSQ r=-0.1	0.1580	0.2381	0.2381	0.2381	0.2381	0.2381
RCSQ r=0.11⋆	0.1675	0.2449	0.2452	0.2479	0.2524	0.2612
RCSQ r=0.3	0.1700	**0.2504**‡	**0.2471**‡	**0.2495**‡	**0.2535**‡	0.2613
RCSQ r=0.5	**0.1710**‡	0.2348	0.2190	0.2291	0.2455	**0.2767**‡
RCSQ r=0.7	0.1630	0.2204	0.2015	0.2262	0.2347	0.2735
RCSQ r=0.9	0.1470	0.1999	0.1771	0.2220	0.2263	0.2764
LSGAIL $\alpha = 0.3$	0.1600	0.2404	**0.2397**	0.2399	0.2403	0.2407
LSGAIL $\alpha = 0.5$	**0.1630**	**0.2410**	0.2346	**0.2405**	0.2408	0.2412
LSGAIL $\alpha = 0.7$	0.1620	0.2406	0.2394	0.2400	**0.2410**	0.2428
LSGAIL $\alpha = 0.9$	0.1600	0.2123	0.1929	0.2066	0.2286	**0.2696**
Oracle	0.2330	0.3233	0.3152	0.3208	0.3298	0.3570



(a) ECRR $\alpha = 0.3$ **(b) ECRR $\alpha = 0.5$** **(c) ECRR $\alpha = 0.7$** **(d) ECRR $\alpha = 0.9$**

Figure 1: Comparing LSGAIL with different α and RCSQ model with different rewards.

indirectly using different evaluation metrics as their training targets for conversation policy. Most of the time, LSGAIL can train the policy to optimize the designated evaluation metric.

Now, we answer three research questions to explain the result table and explain why the results can show our claims in the experiment design are positive.

RQ1: How does different user types (α) affect RCSQ? From Table 2, we can see that none of the RCSQ models give the best performing policy on all the ECRR metrics. The tuned RCSQ with $r = 0.3$ outperforms other models on all the metrics except for Recall@1/100 and ECRR $\alpha = 0.9$. The best performing model on these two metrics is RCSQ with $r = 0.5$. We can also see how the performances of RCSQ with different rewards vary from Fig 1.

The fact that no single universal RCSQ reward can fit all of the user types is why RCSQ is not good enough for conversational search policy learning. We now explain why this result is reasonable. The idea behind the RCSQ model is to model conversational search policy as a risk-controlling task in conversational search actions. (1) The parameter r in RCSQ represents the reward for clarifying questions. Using a small reward makes the policy more conservative about asking clarifying questions, and a higher reward makes the policy more optimistic about asking questions. (2) The parameter α in ECRR reflects the actual degree of conversation action risks in the form of users' patience for clarifying questions. Smaller α represents impatient users, who will be more likely to leave the conversation when seeing bad clarifying questions. Larger α represents patient users, who will spend more time on clarifying questions and interact more with the system.

As a result of (1) and (2), RCSQ with smaller r should theoretically work better on ECRR with smaller α, and worse on ECRR with larger α, and vice versa. which is shown in our experiments (this can be seen from the figure and the cells where RCSQ with $r = 0.5, 0.7, 0.9$ underperform the baselines on ECRR $\alpha = 0.3, 0.5$, while RCSQ with $r = 0.11, 0.3$ underperform on ECRR $\alpha = 0.9$). However, in real-world scenarios, we do not know ahead about the users of our search systems. Even if we do, no reward selection mechanism utilizes our knowledge about the users in RCSQ besides grid search. This makes RCSQ (and all reinforcement learning models) hard to generalize to various user scenarios.

RQ2: How does LSGAIL compare to RCSQ? From Table 2, by comparing the performances of the LSGAIL framework and other

Table 4: An example of LSGAIL choosing the best search system actions.

Turn	Conversation		Analysis
Turn 1	User	(Initial Query) Well i logged out and logged back in and somedude, the new user, still cannot sudo without being told off that this incident will be reported.	This retrieved result is the correct result, but is ranked 6th by the result retrieval model. This clarifying question is relevant, and is ranked 1st by the question retrieval model. In this case, the reward of returning the result is $1/6 = 0.167$, while the reward of asking the question is $r = 0.1 < 0.167$. Because of this, RCSQ chooses to directly return the result to the user. Hence the ECRR of the RCSQ model is equal to $1/6 = 0.167$. LSGAIL chooses to ask with the clarifying question ranked. Let's see what happens in the next turn.
	Result	/etc/sudoerrs is for handling SUDO... I don't know how yours is right now set, but check the content, maybe your default user was autmatically included.	
	Question	Whom password do you use for sudoing?	
Turn 2	User	(Reply) I use the password of the new user, somedude, that i created.	The retrieved result is the same as in turn 1, but it is now ranked 1st by the result retrieval model thanks to more context. The retrieved question is also relevant, but it is ranked 7th by the question retrieval model. LSGAIL chooses to ask no more questions and returns the result, hence ECRR = $0.7^1 * 1 = 0.7$. If the system would ask the question, which is still relevant, and then go to turn 3, it would get an ECRR $= 0.7^7 * 1 \approx 0.08$, which is lower because it would consume the user's time and degrade the user's search experience.
	Result	Same as in turn 1.	
	Question	I see... I was asking because it's a common error using the root's password. Is the user listed in /etc/sudoerrs, or does it belong to the same group than your user?	

policies vertically, we can draw the following conclusions: LSGAIL outperforms all the baseline policies most of the time. Using LSGAIL $\alpha = 0.5$ as an example, LSGAIL outperforms all the baseline policies including Q0A, Q1A, Q2A, and CtxPred on R@1/100, MRR, and ECRR $\alpha = 0.3$ significantly. From the table, the best baseline model is Q0A, with R@1/100 = 0.1580, MRR = 0.2381, and ECRR = 0.2381. LSGAIL gets R@1/100 = 0.1600, MRR = 0.2403, and ECRR = 0.2397, 0.2399, 0.2403. This implies that LSGAIL itself is an effective conversation policy training algorithm.

In general, LSGAIL performances are on par with RCSQ. When compared with RCSQ variations with rewards that are too small or too large, LSGAIL can achive better performances. This is shown in Fig 1. It does perform slightly worse than the finetuned RCSQ model variations with r=0.11 or 0.3 due to the lack of reward signal during training. However, when used on new dataset or unknown user types, LSGAIL does not need to manually finetune to find the best reward to achieve similar performances.

RQ3: How does different user types (α) change the comparison? In RQ2, we see that LSGAIL can keep on par with RCSQ, although it always slightly underperforms the best RCSQ variation. In RQ3, we further study their performances by comparing the result table and Fig 1 horizontally. Surprisingly, we cannot find any single RCSQ reward that always outperforms LSGAIL on all ECRR with different α. Specifically, RCSQ with $r = 0.1, 0.3$ are better than LSGAIL on ECRR with $\alpha = 0.3, 0.5, 0.7$. However, they do not generalize to ECRR with $\alpha = 0.9$, which represents the case with more patient users. To get good performance, RCSQ needs to finetune its reward to the range of $[0.5, 0.9]$. In contrast, LSGAIL never needs reward tuning and achieves comparable performances, regardless of small or large α. This further shows that LSGAIL is theoretically more generalizable and easier to deploy than RCSQ.

6.2 Case Study

We conduct case studies to understand why LSGAIL can outperform the untuned RCSQ model. Table 4 is an example we choose in the experiment that compares LSGAIL $\alpha = 0.9$ and RCSQ $r = 0.1$:

In this example, LSGAIL asks one clarifying question and gets ECRR = 1, RCSQ directly returns the result and gets ECRR = 0.167. From this example, we can see that RCSQ tends to only ask clarifying questions when the retrieved result reciprocal rank is lower than the finetuned reward. Because of this mechanism, RCSQ policy can improve the search result when it is completely irrelevant, e.g., when there are no relevant results in the top 10. However, its shortage is it cannot improve sub-optimal results to good results, e.g., improve the correct result from 6th to 1st. The training of LSGAIL does not set a hard reward for clarifying questions. Hence, it does not have the RCSQ policy problem, and it can further improve sub-optimal results to good results.

7 Conclusions

In this paper, we highlight the necessity of a conversational search policy in conversational search systems. As a solution, we propose a reward-free imitation learning framework for conversational search policy learning to address the problem in reinforcement learning methods. That is, they require heavy reward tuning, and are hard to generalize to different tasks and user types. The reward-free imitation learning framework trains the policy by inferring the best rewards from expert trajectories, which can be computed at a low cost from raw conversational search logs. The algorithm could potentially generalize to any user assumptions. Hence, it solves both of the two problems of reinforcement learning methods.

To show our proposed framework can solve the two problems, we design experiments on the Ubuntu Dialog Corpus dataset and compare our proposed framework with three naive baseline policies, one behavior cloning policy learning method, and one representative reward-tuning reinforcement learning model. To evaluate the entire conversational search trajectory, we propose a new multi-turn evaluation metric called ECRR. Our experiment results show that our proposed framework can work reasonably well without reward tuning, and it can generalize well to different user assumptions. Our paper provides a useful reward-free imitation learning framework for conversational search policy training, which is easier to deploy than traditional reinforcement learning methods and more flexible to various user assumptions.

References

[1] Pieter Abbeel and Andrew Y Ng. 2004. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*. 1.

[2] Mohammad Aliannejadi, Julia Kiseleva, Aleksandr Chuklin, Jeff Dalton, and Mikhail Burtsev. 2020. ConvAI3: Generating clarifying questions for open-domain dialogue systems (ClariQ). *arXiv preprint arXiv:2009.11352* (2020).

[3] Mohammad Aliannejadi, Julia Kiseleva, Aleksandr Chuklin, Jeffrey Dalton, and Mikhail Burtsev. 2021. Building and Evaluating Open-Domain Dialogue Corpora with Clarifying Questions. *arXiv preprint arXiv:2109.05794* (2021).

[4] Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W Bruce Croft. 2019. Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the 42nd international acm sigir conference on research and development in information retrieval*. 475–484.

[5] James Allan. 2005. *HARD track overview in TREC 2003 high accuracy retrieval from documents*. Technical Report. MASSACHUSETTS UNIV AMHERST CENTER FOR INTELLIGENT INFORMATION RETRIEVAL.

[6] Avishek Anand, Lawrence Cavedon, Hideo Joho, Mark Sanderson, and Benno Stein. 2020. Conversational search (dagstuhl seminar 19461). In *Dagstuhl Reports*, Vol. 9. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

[7] Leif Azzopardi, Mateusz Dubiel, Martin Halvey, and Jeffery Dalton. 2018. Conceptualizing agent-human interactions during the conversational search process. In *The second international workshop on conversational approaches to information retrieval*.

[8] Krisztian Balog. 2021. Conversational AI from an Information Retrieval Perspective: Remaining Challenges and a Case for User Simulation. (2021).

[9] Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*. 65–72.

[10] Nir Baram, Oron Anschel, Itai Caspi, and Shie Mannor. 2017. End-to-end differentiable adversarial imitation learning. In *International Conference on Machine Learning*. PMLR, 390–399.

[11] Nicholas J Belkin, Colleen Cool, Adelheit Stein, and Ulrich Thiel. 1995. Cases, scripts, and information-seeking strategies: On the design of interactive information retrieval systems. *Expert systems with applications* 9, 3 (1995), 379–395.

[12] Keping Bi, Qingyao Ai, and W Bruce Croft. 2021. Asking Clarifying Questions Based on Negative Feedback in Conversational Search. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*. 157–166.

[13] Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2016. Learning end-to-end goal-oriented dialog. *arXiv preprint arXiv:1605.07683* (2016).

[14] Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. 2009. Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM conference on Information and knowledge management*. 621–630.

[15] Wenhu Chen, Jianshu Chen, Pengda Qin, Xifeng Yan, and William Yang Wang. 2019. Semantically conditioned dialog response generation via hierarchical disentangled self-attention. *arXiv preprint arXiv:1905.12866* (2019).

[16] Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. *arXiv preprint arXiv:1808.07036* (2018).

[17] W Bruce Croft and Roger H Thompson. 1987. I3R: A new approach to the design of document retrieval systems. *Journal of the american society for information science* 38, 6 (1987), 389–404.

[18] Yashar Deldjoo, Johanne R Trippas, and Hamed Zamani. 2021. Towards multi-modal conversational information seeking. In *Proceedings of the ACM Conference on Research and Development in Information Retrieval, SIGIR*, Vol. 21.

[19] Bhuwan Dhingra, Lihong Li, Xiujun Li, Jianfeng Gao, Yun-Nung Chen, Faisal Ahmed, and Li Deng. 2016. Towards end-to-end reinforcement learning of dialogue agents for information access. *arXiv preprint arXiv:1609.00777* (2016).

[20] Mehdi Fatemi, Layla El Asri, Hannes Schulz, Jing He, and Kaheer Suleman. 2016. Policy networks with two-stage training for dialogue systems. *arXiv preprint arXiv:1606.03152* (2016).

[21] Chelsea Finn, Paul Christiano, Pieter Abbeel, and Sergey Levine. 2016. A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models. *arXiv preprint arXiv:1611.03852* (2016).

[22] Zuohui Fu, Yikun Xian, Yongfeng Zhang, and Yi Zhang. 2020. Tutorial on Conversational Recommendation Systems. In *Fourteenth ACM Conference on Recommender Systems*. 751–753.

[23] Jianfeng Gao, Michel Galley, and Lihong Li. 2018. Neural approaches to conversational ai. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1371–1374.

[24] Jianfeng Gao, Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Heung-Yeung Shum. 2020. Robust conversational AI with grounded text generation. *arXiv preprint arXiv:2009.03457* (2020).

[25] Jianfeng Gao, Chenyan Xiong, Paul Bennett, and Nick Craswell. 2022. Neural Approaches to Conversational Information Retrieval. *arXiv preprint arXiv:2201.05176* (2022).

[26] Sarik Ghazarian, Johnny Tian-Zheng Wei, Aram Galstyan, and Nanyun Peng. 2019. Better automatic evaluation of open-domain dialogue systems with contextualized embeddings. *arXiv preprint arXiv:1904.10635* (2019).

[27] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems* 27 (2014).

[28] Sanda Harabagiu, Andrew Hickl, John Lehmann, and Dan Moldovan. 2005. Experiments with interactive question-answering. In *Proceedings of the 43rd annual meeting of the association for computational linguistics (ACL'05)*. 205–214.

[29] Claudia Hauff, Julia Kiseleva, Mark Sanderson, Hamed Zamani, and Yongfeng Zhang. 2021. Conversational Search and Recommendation: Introduction to the Special Issue.

[30] Matthew Henderson, Paweł Budzianowski, Inigo Casanueva, Sam Coope, Daniela Gerz, Girish Kumar, Nikola Mrkšić, Georgios Spithourakis, Pei-Hao Su, Ivan Vulić, et al. 2019. A repository of conversational datasets. *arXiv preprint arXiv:1904.06472* (2019).

[31] Jonathan Ho and Stefano Ermon. 2016. Generative adversarial imitation learning. *Advances in neural information processing systems* 29 (2016), 4565–4573.

[32] Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2019. Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring. *arXiv preprint arXiv:1905.01969* (2019).

[33] Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. 2017. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)* 50, 2 (2017), 1–35.

[34] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

[35] Daniel Jurafsky and James H Martin. [n.d.]. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.

[36] Abhishek Kaushik and Gareth JF Jones. 2021. A Conceptual Framework for Implicit Evaluation of Conversational Search Interfaces. *arXiv preprint arXiv:2104.03940* (2021).

[37] Diane Kelly, Paul B Kantor, Emile L Morse, Jean Scholtz, and Ying Sun. 2009. Questionnaires for eliciting evaluation data from users of interactive question answering systems. *Natural Language Engineering* 15, 1 (2009), 119–141.

[38] Johannes Kiesel, Arefeh Bahrami, Benno Stein, Avishek Anand, and Matthias Hagen. 2018. Toward voice query clarification. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1257–1260.

[39] Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1437–1447.

[40] Aldo Lipani, Ben Carterette, and Emine Yilmaz. 2019. From a user model for query sessions to session rank biased precision (sRBP). In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*. 109–116.

[41] ALDO LIPANI, BEN CARTERETTE, and EMINE YILMAZ. 2021. How Am I Doing?: Evaluating Conversational Search Systems Offline. *ACM Transactions on Information Systems* (2021).

[42] Mengyang Liu, Yiqun Liu, Jiaxin Mao, Cheng Luo, and Shaoping Ma. 2018. Towards designing better session search evaluation metrics. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1121–1124.

[43] Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. *arXiv preprint arXiv:1506.08909* (2015).

[44] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. 2017. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*. 2794–2802.

[45] Alistair Moffat and Justin Zobel. 2008. Rank-biased precision for measurement of retrieval effectiveness. *ACM Transactions on Information Systems (TOIS)* 27, 1 (2008), 1–27.

[46] Nikola Mrkšić, Diarmuid O Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2016. Neural belief tracker: Data-driven dialogue state tracking. *arXiv preprint arXiv:1606.03777* (2016).

[47] Robert N Oddy. 1977. Information retrieval through man-machine dialogue. *Journal of documentation* (1977).

[48] 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*. 311–318.

[49] Baolin Peng, Xiujun Li, Lihong Li, Jianfeng Gao, Asli Celikyilmaz, Sungjin Lee, and Kam-Fai Wong. 2017. Composite task-completion dialogue policy learning via hierarchical deep reinforcement learning. *arXiv preprint arXiv:1704.03084* (2017).

[50] Gustavo Penha, Alexandru Balan, and Claudia Hauff. 2019. Introducing MANtIS: a novel multi-domain information seeking dialogues dataset. *arXiv preprint arXiv:1912.04639* (2019).

[51] Gustavo Penha and Claudia Hauff. 2020. Challenges in the Evaluation of Conversational Search Systems.. In *Converse@ KDD*.

[52] D Pomerleau. 1998. An autonomous land vehicle in a neural network. *Advances in Neural Information Processing Systems; Morgan Kaufmann Publishers Inc.: Burlington, MA, USA* (1998).

[53] C. Qu, L. Yang, W. B. Croft, J. Trippas, Y. Zhang, and M. Qiu. 2018. Analyzing and Characterizing User Intent in Information-seeking Conversations.. In *SIGIR '18*.

[54] C. Qu, L. Yang, W. B. Croft, Y. Zhang, J. Trippas, and M. Qiu. 2019. User Intent Prediction in Information-seeking Conversations. In *CHIIR '19*.

[55] Filip Radlinski and Nick Craswell. 2017. A theoretical framework for conversational search. In *Proceedings of the 2017 conference on conference human information interaction and retrieval*. 117–126.

[56] Sudha Rao and Hal Daumé III. 2019. Answer-based adversarial training for generating clarification questions. *arXiv preprint arXiv:1904.02281* (2019).

[57] Jiaming Song, Hongyu Ren, Dorsa Sadigh, and Stefano Ermon. 2018. Multi-agent generative adversarial imitation learning. *arXiv preprint arXiv:1807.09936* (2018).

[58] Shang-Yu Su, Xiujun Li, Jianfeng Gao, Jingjing Liu, and Yun-Nung Chen. 2018. Discriminative deep dyna-q: Robust planning for dialogue policy learning. *arXiv preprint arXiv:1808.09442* (2018).

[59] Ryuichi Takanobu, Hanlin Zhu, and Minlie Huang. 2019. Guided dialog policy learning: Reward estimation for multi-domain task-oriented dialog. *arXiv preprint arXiv:1908.10719* (2019).

[60] Faraz Torabi, Garrett Warnell, and Peter Stone. 2018. Generative adversarial imitation from observation. *arXiv preprint arXiv:1807.06158* (2018).

[61] Svitlana Vakulenko. 2019. Knowledge-based Conversational Search. *arXiv preprint arXiv:1912.06859* (2019).

[62] Zhenduo Wang and Qingyao Ai. 2021. Controlling the Risk of Conversational Search via Reinforcement Learning. In *Proceedings of the Web Conference 2021*. 1968–1977.

[63] Zhenduo Wang and Qingyao Ai. 2022. Simulating and Modeling the Risk of Conversational Search. *arXiv preprint arXiv:2201.00235* (2022).

[64] Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2016. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562* (2016).

[65] Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. *arXiv preprint arXiv:1905.08743* (2019).

[66] Markus Wulfmeier, Peter Ondruska, and Ingmar Posner. 2015. Deep inverse reinforcement learning. *CoRR, abs/1507.04888* (2015).

[67] Markus Wulfmeier, Peter Ondruska, and Ingmar Posner. 2015. Maximum entropy deep inverse reinforcement learning. *arXiv preprint arXiv:1507.04888* (2015).

[68] Jingjing Xu, Yuechen Wang, Duyu Tang, Nan Duan, Pengcheng Yang, Qi Zeng, Ming Zhou, and Xu Sun. 2019. Asking clarification questions in knowledge-based question answering. 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)*. 1618–1629.

[69] Liu Yang, Minghui Qiu, Chen Qu, Cen Chen, Jiafeng Guo, Yongfeng Zhang, W Bruce Croft, and Haiqing Chen. 2020. IART: Intent-aware response ranking with transformers in information-seeking conversation systems. In *Proceedings of The Web Conference 2020*. 2592–2598.

[70] L. Yang, M. Qiu, C. Qu, J. Guo, Y. Zhang, W. B. Croft, J. Huang, and H. Chen. 2018. Response Ranking with Deep Matching Networks and External Knowledge in Information-seeking Conversation Systems. In *SIGIR '18*.

[71] 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*. 1933–1936.

[72] Shi Yu, Zhenghao Liu, Chenyan Xiong, Tao Feng, and Zhiyuan Liu. 2021. Few-Shot Conversational Dense Retrieval. *arXiv preprint arXiv:2105.04166* (2021).

[73] Munazza Zaib, Wei Emma Zhang, Quan Z Sheng, Adnan Mahmood, and Yang Zhang. 2021. Conversational Question Answering: A Survey. *arXiv preprint arXiv:2106.00874* (2021).

[74] Hamed Zamani and Nick Craswell. 2020. Macaw: An extensible conversational information seeking platform. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2193–2196.

[75] Hamed Zamani, Susan Dumais, Nick Craswell, Paul Bennett, and Gord Lueck. 2020. Generating clarifying questions for information retrieval. In *Proceedings of The Web Conference 2020*. 418–428.

[76] Hamed Zamani, Gord Lueck, Everest Chen, Rodolfo Quispe, Flint Luu, and Nick Craswell. 2020. Mimics: A large-scale data collection for search clarification. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 3189–3196.

[77] Hamed Zamani, Bhaskar Mitra, Everest Chen, Gord Lueck, Fernando Diaz, Paul N Bennett, Nick Craswell, and Susan T Dumais. 2020. Analyzing and learning from user interactions for search clarification. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1181–1190.

[78] Hamed Zamani, Johanne R Trippas, Jeff Dalton, and Filip Radlinski. 2022. Conversational Information Seeking. *arXiv preprint arXiv:2201.08808* (2022).

[79] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675* (2019).

[80] Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W Bruce Croft. 2018. Towards conversational search and recommendation: System ask, user respond. In *Proceedings of the 27th acm international conference on information and knowledge management*. 177–186.

[81] Zhuosheng Zhang, Jiangtong Li, Pengfei Zhu, Hai Zhao, and Gongshen Liu. 2018. Modeling multi-turn conversation with deep utterance aggregation. *arXiv preprint arXiv:1806.09102* (2018).

[82] Zheng Zhang, Ryuichi Takanobu, Qi Zhu, MinLie Huang, and XiaoYan Zhu. 2020. Recent advances and challenges in task-oriented dialog systems. *Science China Technological Sciences* 63, 10 (2020), 2011–2027.

[83] Tiancheng Zhao, Kaige Xie, and Maxine Eskenazi. 2019. Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models. *arXiv preprint arXiv:1902.08858* (2019).

[84] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. 2008. Maximum entropy inverse reinforcement learning.. In *Aaai*, Vol. 8. Chicago, IL, USA, 1433–1438.