Evaluating Conversational Recommender Systems via User Simulation

Shuo Zhang*
Bloomberg
London, United Kingdom
szhang611@bloomberg.net

Krisztian Balog
University of Stavanger
Stavanger, Norway
krisztian.balog@uis.no

ABSTRACT

Conversational information access is an emerging research area. Currently, human evaluation is used for end-to-end system evaluation, which is both very time and resource intensive at scale, and thus becomes a bottleneck of progress. As an alternative, we propose automated evaluation by means of simulating users. Our user simulator aims to generate responses that a real human would give by considering both individual preferences and the general flow of interaction with the system. We evaluate our simulation approach on an item recommendation task by comparing three existing conversational recommender systems. We show that preference modeling and task-specific interaction models both contribute to more realistic simulations, and can help achieve high correlation between automatic evaluation measures and manual human assessments.

CCS CONCEPTS

• **Information systems** → **Users and interactive retrieval**; **Recommender systems**; • **Human-centered computing** → *HCI design and evaluation methods*;

KEYWORDS

User simulation; conversational recommendation; conversational information access

ACM Reference Format:
Shuo Zhang and Krisztian Balog. 2020. Evaluating Conversational Recommender Systems via User Simulation. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20), August 23–27, 2020, Virtual Event, CA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394486.3403202

1 INTRODUCTION

Conversational information access is a newly emerging research area that aims at providing access to digitally stored information over a dialog interface [33, 34]. It is specifically concerned with a goal-oriented sequence of exchanges, including complex information seeking and exploratory information gathering, and multi-step task completion and recommendation [10]. In this paper, we focus on the problem of item recommendation. The conversational paradigm is particularly suited for this task, as it allows people to disclose their preferences [8], efficiently explore the search space [37], and provide fine-grained feedback [6]. More specifically, we address the problem of evaluating conversational recommender systems (referred to as *conversational agents*).

Test-collection based evaluation has a large history in information retrieval (IR) [28], but it has limitations. It is possible to create an offline test collection for conversational agents to select the best response, in answer to a user utterance, from a set of possible candidates. Assuming that the candidate generation step has been addressed by a different component, such reusable test collection would enable the comparison of different response ranking methods. This is exactly the approach taken by the TREC Conversational Assistance Track benchmark initiative [11] and also by others [1]. However, this assessment is limited in scope to a single turn in a conversation; it does not tell us anything about the overall usefulness of the system or about users' satisfaction with the flow of the dialogue. Collecting and annotating entire conversations is an option, but it is expensive, time-consuming, and does not scale. Importantly, it would not yield a reusable test collection. The evaluation of conversational information access systems, therefore, represents an open challenge and calls for additional methodologies to be considered. Possible alternatives include laboratory user studies [16], online evaluation [14], and simulated users [22]. Of these, we will be exploring user simulation in this work.

Our objective is to develop a user simulator that is (1) capable of producing responses that a real user would give in a certain dialog situation [31], and (2) would enable to compute an automatic assessment of a conversational agent such that it is predictive of its performance with real users. We wish to accomplish this without making specific assumptions about the inner workings of conversational agents. That is, we treat them much like black boxes. We further wish the simulator to be data driven, such that it can be used with any conversational agent only by supplying a small corpus of annotated dialogues real users have conducted with the agent.

We build on the well-established Agenda-based User Simulator [30] as our general framework, and explore multiple options for modeling interactions and user preferences. Specifically, we develop a task-specific interaction model to more directly capture the flow of the conversational item recommendation task. For preference modeling, we present an approach for ensuring the consistency of responses based on the notion of personal knowledge graphs [4]. We evaluate our simulation approaches by comparing three existing conversational movie recommender systems, using both automatic and manual evaluation. We find that more sophisticated interaction

*Work done while at the University of Stavanger, Norway.

KDD '20, August 23–27, 2020, Virtual Event, CA, USA

ACM ISBN 978-1-4503-7998-4/20/08…$15.00
https://doi.org/10.1145/3394486.3403202

and preference modeling leads to more realistic simulation, and we achieve high overall correlation with real users.

In summary, this paper makes the following novel contributions:

- A general framework for evaluating conversational recommender agents via simulation.
- Interaction and preference models to better control the conversation flow and to ensure the consistency of responses given by the simulated user.
- An experimental comparison of three conversational movie recommender agents, using both real and simulated users.
- An analysis of comments collected from human evaluation, and identification of areas for future development.

Our simulation platform is made publicly available.[1]

2 RELATED WORK

Our work investigates how to evaluate conversational recommender systems via user simulation, and is located in the intersection dialogue systems, conversational information access, and evaluation.

2.1 Dialogue Systems

Dialogue systems communicate with users in natural language (text, speech, or both). They can be broadly categorized into two groups: *non-task-oriented systems* (also known as *chatbots*) and *task-oriented systems* [7, 32]. Chatbots aim to carry on an extended conversation ("chit-chat") with the goal of mimicking unstructured human-human interactions. Task-oriented systems, on the other hand, aim to assist users to complete some specific task (e.g., give navigation directions, control appliances, book a flight, buy a product, etc.). Our work falls in this latter category. Modern task-oriented dialogue systems are based on a *dialogue-state* (or *belief-state*) architecture [15], capitalizing on the notion of *dialogue acts* (i.e., task-specific intents that are being communicated).

There is a long history of utilizing user simulation in the context of spoken dialog systems [31]. Simulation is mainly used for dialogue policy learning and end-to-end dialogue training, in order to reduce time and effort by generating large-scale utterances of real users [31]. Early work can be categorized into rule-based [9] and corpus-based methods [12, 30]. Recent works employ neural approaches, esp. sequence-to-sequence models [2, 17]. The most widely used approach for policy optimization is the Agenda-Based User Simulator [30], which represents the user state as a stack of user actions, called the agenda. Our work also builds on this method. Simulation can also be used to evaluate different aspects of a dialogue system [12], which is our focus in this paper.

2.2 Conversational Information Access

Conversational information access is concerned with a goal-oriented sequence of exchanges [10], where the agent aims to help the user to satisfy their information need, by supporting them in finding, exploring, and understanding the possible options and information objects that are available [3]. When resolving information needs, the conversational agent should consider both short- and long-term knowledge of the user [27]. Recently, progress has been made on specific subtasks for conversational information access, including response ranking [36], asking clarifying questions [1], predicting user intent [26], and preference elicitation [6, 8]. End-to-end evaluation, however, has received little attention to date, due to the lack of appropriate evaluation resources and methodology. With this paper, we aim to start filling this gap.

2.3 Evaluation

Conversational recommenders follow a task-oriented dialog system architecture, consisting of natural language understanding (NLU), natural language generation (NLG), and dialog manager (DM). Evaluation may be performed on the component-level or end-to-end.

Component-level evaluation has primarily focused on NLU and NLG. NLU is often viewed as a classification task and is evaluated in terms of precision, recall, and F1-score [2, 23] or intent/slot error rates [20]. NLG is commonly assessed using word overlap-based metrics from machine translation, such as BLEU, METEOR, and ROUGE [5, 23]. These metrics, however, turn out to correlate very poorly with human judgments, due to the many possible responses to any given turn [21]. An alternative is to consider the meaning of each word by using embedding-based metrics [21]. In addition to automatic means of evaluation, where all the above metrics fail, human evaluation is also considered. For example, Belz and Reiter [5] use NIST, BLEU, and ROUGE for automatic evaluation, and a 6-point scale for human evaluation. They find that the automatic metrics can be expected to correlate well with human judgments only if the reference texts used are of high quality.

End-to-end evaluation assesses the dialogue quality based on the generated dialogues. Metrics include but not limited to success rate, reward, and average dialogue turns [23, 24]. We also use these metrics in our evaluation. Human evaluation has also been performed in terms of success rate [24] and slot errors [20]. A recently proposed alternative is *adversarial evaluation* [19]. Inspired by the Turing test, a classifier is trained to distinguish between human-generated and machine-generated responses; the more successful a system is at "fooling" the classifier, the better it is. We perform a similar evaluation, but we ask crowd workers to try to perform this classification between real and simulated users.

3 PROBLEM STATEMENT

Our goal is to develop an approach for evaluating conversational recommender systems (agents) via simulated users. We specify two main requirements for the user simulator. First, is should be capable of producing responses that a real user would give in a certain dialog situation [31]. Specifically, this entails (R1) generating responses that are *consistent* with users' *preferences*, and (R2) being able to *follow* a task specific *dialog flow*. Second, the simulator should (R3) enable to compute an *automatic assessment* of the agent such that it is *predictive* of its performance with real users. Formally:

> For a given system S and user population U, the goal of user simulation U^* is to predict the performance of S when used by U, denoted as $M(S, U)$. For two systems S_1 and S_2, U^* should be such that if $M(S_1, U) < M(S_2, U)$ then $M(S_1, U^*) < M(S_2, U^*)$.

3.1 Simulation Framework

The user simulator consists of the following main components, which are illustrated in Fig. 1: natural language understanding, response generation, and natural language generation. Our main focus in this paper is on the response generation part.

[1] https://github.com/iai-group/kdd2020-usersim

Table 1: Agent and user actions considered in this paper. Main actions are boldfaced.

Category	Agent	User
Query Formulation	**Reveal**: Disclose, Non-disclose, Revise, Refine, Expand	**Inquire**: Elicit, Clarify
Set Retrieval	**Inquire**: List, Compare, Subset, Similar, **Navigate**: Repeat, Back, More, Note, Complete	**Reveal**: Show, List, Similar, Subset **Traverse**: Repeat, Back, More, Record, End
Mixed Initiative	**Interrupt**, **Interrogate**	**Suggest**

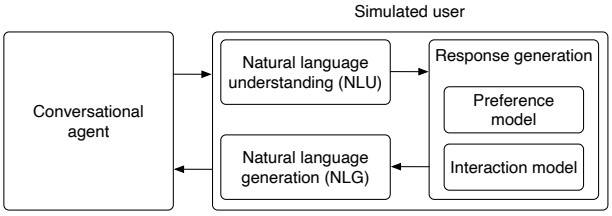

Figure 1: Architecture of our user simulator.

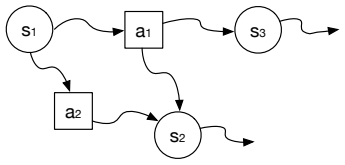

Figure 2: Dialogue as a Markov Decision Process [31].

Natural language understanding is the task of translating an agent utterance into a structured format. We will assume that the user has the capacity to perfectly understand the intent behind the agent's utterances. *Response generation* is concerned with determining the next user action based on the understanding of the system's utterance. To ensure the consistency of user preferences (R1), we employ a *preference model*, which is a structured representation of item- and set-level user preferences, based on the notion of a personal knowledge graph. To follow a task-specific dialog flow (R2), we utilize an *interaction model*. We will assume that the user has some expectations regarding how the agent should act (i.e., we impose a pre-defined interaction policy). *Natural language generation* is the process of turning a structured response representation into natural language. We will use a simple template-based approach to turn structured intent representations into natural language text.

4 MODELING SIMULATED USERS

Our objective is to simulate users for a specific task: conversational item recommendation. Modeling dialogue as a Markov Decision Process, we employ agenda-based simulation [30] as the overall simulation framework (Sect. 4.1). It operates on the notion of *dialogue acts*, referred to as *actions* henceforth, which represent task-specific intents that are being communicated in utterances. Regarding the choice of actions, we take a subset of actions identified in [3] and list them in Table 1. To operationalize the agenda-based framework, we need a model of interaction that guides the simulated user through the conversation, i.e., helps to determine how to respond. We consider both an existing general-purpose model and introduce a task-specific alternative (Sect. 4.2). Furthermore, we need to model the preferences of the simulated user. We present two alternatives, both of which hinge on the idea of sampling from historical user-item interactions (Sect. 4.3). Noting that this is not our focus, we detail natural language understanding and generation in Sect. 4.4. Overall, our approach does not make any assumptions about the inner workings of the recommender agent. We, however, assume that a small corpus of dialogs between humans and the agent, with turns annotated with user/agent actions, is available for training various components of the simulator.

4.1 Agenda-based Simulation

Dialogue can effectively be modeled as a Markov Decision Process (MDP) [31]. Every MDP is formally described by a finite space $\mathcal{S}$, a finite action set $\mathcal{A}$, and a set of transition probabilities. At each time step (dialogue turn) t, the dialogue manager is in a particular state $s_t \in \mathcal{S}$. By executing action $a_t \in \mathcal{A}$, it transitions into the next state s_{t+1} according to the transition probability $P(s_{t+1}|s_t, a_t)$. The Markov property ensures that the state at time $t+1$ depends only on the state and action at time t:

$$P(s_{t+1}|s_t, a_t, s_{t-1}, a_{t-1}, \ldots, s_0, a_0) = P(s_{t+1}|s_t, a_t) \; .$$

Using the MDP model of dialogue, the dialogue manager can be visualized as an agent traveling through a network of dialogue states; see Fig. 2.

The agenda-based simulator [30] provides a probabilistic method for bootstrapping the MDP dialogue process. The user state s is factorized into action agenda A and information-seeking goal g. A is a stack-like representation for user actions that is dynamically updated; the next user action is selected from the top of the agenda. Specifically, the user agenda A is a stack-like structure of length n, containing the user dialogue actions, where $A[1]$ denotes the bottom and $A[n]$ denotes the top item. The update of A is coherent to the state; new actions are pushed onto the agenda, and no longer relevant ones are removed. We use s_t and s_{t+1} to represent two consecutive states in the diagram, and a_t to represent the user action selected from A_t. A_{t+1} is the agenda after taking a_t, and it is derived from A_t for simplification. This way we have formalized the conversation into a sequence of states; Fig. 3 illustrates state transitions via an example. Accordingly, the estimation of the probability of going from one state to the next goes as follows:

$$P(s_{t+1}|s_t, A_t) = P(A_{t+1}|A_t, g_{t+1}) \cdot P(g_{t+1}|A_t, g_t),$$

where $P(A_{t+1}|A_t, g_{t+1})$ is the agenda update and $P(g_{t+1}|A_t, g_t)$ is the goal update. Goal g is further decomposed into constrains C, specifying the type of information sought, and requests R, which specify the additional pieces of information requested from the agent. We construct g based on the preference model (to be detailed in Sect. 4.2). The goal update is formalized as follows:

$$P(g_{t+1}|A_t, g_t) = P(R_{t+1}|A_t, R_t, C_{t+1}) \cdot P(C_{t+1}|A_t, R_t, C_t) \cdot \delta(g_{t+1}, g_t).$$

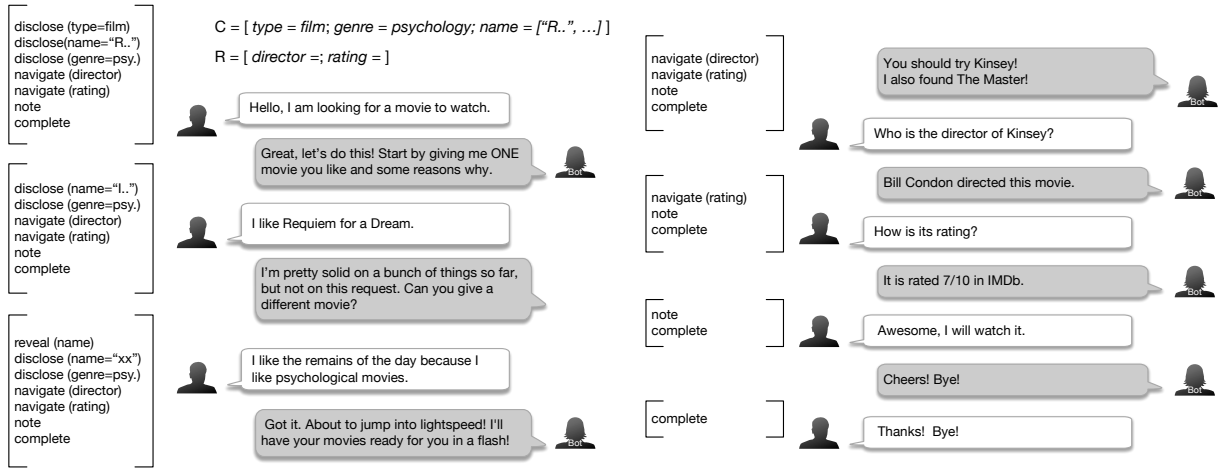

Figure 3: Example dialogue with agenda sequence and state transition. The agenda is shown in square brackets. The third agenda is a result of a push operations, all other agendas updates are pull operations.

As suggested by Schatzmann et al. [30], the goal update may be simplified by hand-crafted heuristics. Our heuristic is to check whether the agent understands the user action and gives the corresponding response. Thus, we base it only on an indicator function δ:

$$P(g_{t+1}|A_t, g_t) = \delta(g_{t+1}, g_t) \,,$$

where $\delta(g_{t+1}, g_t)$ returns 1 if the goal g_t was accomplished and otherwise returns 0.

Agenda updates are regarded as a sequence of pull or push operations, where dialogue actions are removed from or added to the top. An accomplished goal ($\delta(g_{t+1}, g_t) = 1$) indicates a pull, otherwise push. For pull, the state transition probability simplifies to:

$$P(s_{t+1}|A_t, s_t) = P(A_{t+1}|A_t, g_{t+1}) \,. \tag{1}$$

For the push operation, we need to find a replacement action $\widetilde{a}_t$, which remains to have the same goal as the original action a_t. The state transition probabilities are then computed according to:

$$P(A_{t+1}|A_t, g_{t+1}) = P(\widetilde{a}_t|A_t, g_{t+1}) \,. \tag{2}$$

The agenda updates, namely, the pull operation ($P(A_{t+1}|A_t, g_{t+1})$) and finding the replacement action in case of a push operation ($P(\widetilde{a}_t|A_t, g_{t+1})$) are informed by the interaction model, and will be detailed in the next subsection.

To sum up, we switch between pull and push (replace) operations by checking if the user action is met with an appropriate agent response. The dialogue is terminated when the agenda is empty.

4.2 Interaction Model

The interaction model defines how the agenda should be initialized (A_0) and updated ($A_t \rightarrow A_{t+1}$) throughout the conversation. We consider two interaction models: (1) an existing general-purpose conversational interaction model, QRFA, which will serve as our baseline, and (2) our model, CIR6, which is developed specifically for the conversational item recommendation task. Before we detail these models, we need to specify the space of possible user actions.

4.2.1 Action Space. We base our user actions $\mathcal{A}$ on agent-human interactions for conversational search by Azzopardi et al. [3], which are listed below (with examples taken from [3]).

- **Disclose**: The user expresses the information need either actively, or in response to the agent's question ("*I would to arrange a holiday in Italy.*").
- **Reveal**: It refers to the user revising, refining, or expanding constraints and requirements ("*Actually, we need to go on the 3rd of May in the evening.*" or "*Can you also check to see what kind of holidays are there available in Spain?*").
- **Inquire**: Once the agents starts to show recommendations, the user may ask for related items ("*Tell me about all the different things you can do in this place.*") or ask for similar options ("*What other regions in Europe are like that?*").
- **Navigate**: In our definition, navigation entails both actions around navigating a list of recommendations ("*Which one is the cheapest option?*") as well as questions about a certain recommended item on the list ("*What's the price of that hotel?*").
- **Note**: During the conversation, the user could mark or save specific items ("*That hotel could be a possibility.*" or "*Save that hotel for later.*").
- **Complete**: Finally, the user can mark the end of the conversation ("*Thanks for the help, bye.*").

Note that we only use user actions to compose the agenda. That is, we generate the next action in the agenda directly based on the current user action, while treating the agent much like a black box. We assume, however, that the simulator can "understand" a set of agent actions. Specifically, we consider the agent actions listed in Table 1 (for a detailed description of each, we refer the reader to [3]). The NLU is trained to recognize this set of agent actions. Then, at each turn, the agenda-based simulator can determine whether the agent responds to the user with an appropriate action (as captured by the indicator function δ). For example, an *Inquire* user action can accept *List* or *Elicit* as an agent response; the full mapping is excluded due to space constraints and will be made available online.

4.2.2 QRFA Model. QRFA (Query, Request, Feedback, and Accept) [35] is a general model for conversational information seeking processes. It uses a simple schema for annotating utterances, with four basic classes: two for user (*Query* and *Feedback*) and two for

Table 2: Mapping the action set used in this paper to high-level QRFA categories.

Category	Actions
Query	Reveal, Disclose, Non-disclose, Revise, Refine, Expand, Inquire, List, Compare, Subset, Similar, Navigate, Repeat, Interrupt, Interrogate
Request	Inquire, Elicit, Clarify, Suggest
Feedback	Back, More, Note, Complete
Answer	Show, List, Similar, Subset, Repeat, Back, More, Record, End

agent (*Request* and *Answer*); see Fig. 4. Vakulenko et al. [35] use this model to discover frequent sequence patterns in dialogs with the help of process mining techniques. QRFA provides good flexibility and generalizability to a wide number of use cases. However, we need to make some adjustments before it can be applied in our scenario. First, for simulation purposes, where we are only interested in the user side, which has only two high-level classes (*Query* and *Feedback*). We subdivide the high-level QRFA categories into our more fine-grained set of actions, as shown in Table 2. Second, agenda initialization is a reverse process to pattern discovery, and there is a lack of methods. Therefore, we take an initial agenda A_0 by sampling from an annotated training corpus of human-agent conversations. When estimating the state transition probabilities, we leverage the agent action b_t (which is either *Request* or *Action*) that happens between two consecutive user actions a_t and a_{t+1} as a two-step transition probability. The transition probability matrix is estimated based on the training corpus. The transition probability between actions is then defined as follows:

$$P(A_{t+1}|A_t, g_{t+1}) = \begin{cases} P(A_{t+1}|b_t)P(b_t|A_t) & \delta(g_{t+1}, g_t) = 1 \\ P(\widetilde{a}_t|b_t) & \delta(g_{t+1}, g_t) = 0, \end{cases}$$

where $\delta(g_{t+1}, g_t)$ indicates whether b_t responds to a_t with an appropriate action. If yes, then we perform a pull operation and remove a_t from the agenda. Otherwise, it is a push operation, where a replacement action $\widetilde{a}_t$ is sampled based on the last agent action b_t. Mind that the agenda updates are performed on the course-grained level (i.e., only Query and Feedback actions). We then probabilistically sample a corresponding fine-grained (sub)action (cf. Table 2) based on historical dialogs.

4.2.3 CIR6 Model. Next, we present our interaction model, which is designed to more directly capture the flow of the conversational item recommendation task. It considers six main user action, hence it is termed CIR6 (for Conversational Item Recommendation). Figure 5 presents the state diagram. Using this model, we can generate the next action in the agenda directly based on the current user action, without having to resort to transition probability estimations for agent actions. Formally:

$$P(A_{t+1}|A_t, g_{t+1}) = \begin{cases} P(A_{t+1}|A_t) \cdot \mathbb{1}(a_{t+1}, a_t) & \delta(g_{t+1}, g_t) = 1 \\ P(\widetilde{a}_t|b_t) & \delta(g_{t+1}, g_t) = 0, \end{cases}$$

where $\mathbb{1}(a_{t+1}, a_t)$ indicates if the two consecutive actions are connected in the state diagram (Fig. 5) or not. We compute the conditional probability $P(A_{t+1}|A_t)$ based on action distributions in a training corpus, i.e., the number of times a_t was followed by a_{t+1}.

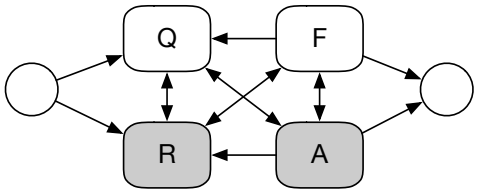

Figure 4: The QRFA model [35]. Agent actions are in grey.

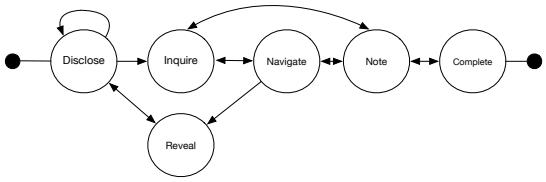

Figure 5: State diagram of our CIR6 model. Different from QRFA, we model only user actions as states.

4.3 Preference Model

The preference model is meant to capture individual differences and personal tastes. Here, preferences are represented as a set of attribute-value pairs. We assume that a sufficiently large corpus of historical user-item interactions is available. In order to create a realistic preference model, we first randomly choose a user from the corpus, then subsample from historical interactions of that user. The set of sampled items is denoted as I_u. We will assume that the simulated user has seen/consumed all items in this set.

4.3.1 Single Item Preference. Recommender systems mostly elicit preferences by asking the user to provide one or more favored items (e.g., favorite movies). Whenever a user is prompted whether they had seen/consumed a specific item i, we check if $i \in I_u$ an answer accordingly. When the user is prompted for their preference of a given item (e.g., "*Did you like it?*") we provide a positive/negative response by flipping a coin. This approach, therefore, offers limited consistency. Items that are seen/consumed are rooted in real user behavior, but the preferences expressed about them are not.

4.3.2 Personal Knowledge Graph. In order to have a more realistic model of user preferences, we build a personal knowledge graph (PKG) [4]. The PKG has two types of nodes: items and attributes. For this approach, we will assume that the corpus of historical user-item interactions contains not only seen/consumed information, but also preferences (i.e., ratings). We divide the I_u into sets of liked and disliked items, I_u^+ and I_u^-, respectively, based on the ratings. Given an attribute $j \in J$, we infer the rating for that attribute by considering the ratings of items that have that attribute:

$$r_j = \frac{1}{|I_j|} \sum_{i \in I_j} r_i,$$

where I_j denotes the set of items that have attribute j, and r_i is the rating of the item i ($r_i \in [-1, 1]$). We can then classify attributes into liked and disliked sets, J_u^+ and J_u^-, respectively. Whenever the user is asked about preferences of a specific item or attribute, those answers are based on the PKG. This ensures that all preference statements expressed by the simulated user will be consistent.

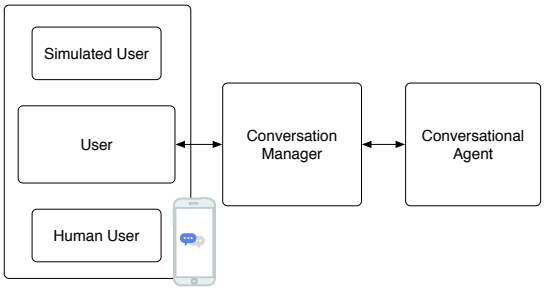

Figure 6: Architecture of our evaluation platform.

4.4 NL Understanding and Generation

Natural language understanding (NLU) is responsible for annotating agent utterances with actions (according to Table 1) and entities (by linking them to a domain specific knowledge base). To carry out these tasks, we assume that a small corpus of dialogs with the conversational agent is available, which is labeled with agent actions. The size of the annotated corpus depends on the variety of language the agent uses. For example, the agents we will consider in our evaluation use rather rigid patterns, therefore, only a limited amount of labeled data is required. We use a simple retrieval-based approach for NLU, where we identify the most similar utterance from the corpus for each input utterance, and take the corresponding action [15]. For entity linking, we first extract patterns (templates) from the labeled corpus that contain placeholders for entity mentions. Then, we use a retrieval-based approach for disambiguation based on surface forms alone.

Natural language generation (NLG) is concerned with creating a textual user utterance based on the user action and associated slots. We follow a template-based approach where, for each action, we randomly select from a small number of hand-crafted response variations. To make it more human-like, a few of the templates purposefully contain typos.

5 EVALUATION ARCHITECTURE

We evaluate conversational agents with both real and simulated users. This is facilitated by a conversation manager, which is a glue module connecting the agent and simulated/real users; see Fig. 6.

5.1 Conversational Agents

We consider three conversational agents for movie recommendation; two of these are existing third-party systems, while the third one was developed by us.

- *And chill*[2] is a single-purpose, consumer-oriented chatbot that a user can send messages to on Facebook and ask for a Netflix recommendation. After answering a few questions such as a liked movie and the reason why liking it, the agent sends movie recommendations based on the user's preferences.
- *Kelly Movie Bot*[3] is a simple bot that answers questions about a specific movie, such as rating, genre, and can also recommend similar movies. The underlying data collection is the Kaggle Movies Recommender System dataset,[4] which is based on the MovieLens dataset. The natural language components utilize the IBM Watson API services.[5] We extended the original Kelly Movie Bot with a number of additional intents (to allow users to indicate their preferences and whether they have already watched a given movie).
- *Our movie recommender system* is based on the Plato Research Dialogue System.[6] This agent can answer questions about movies (such as directors, summary, ratings, etc.) and can provide recommendations based on genres. It also solicits feedback on movies the user has watched.

For our experiments, we anonymized the three agents, by assigning labels **A**, **B**, and **C** to them in random order. All three agents support query formulation, set retrieval, and mixed-initiative properties. For two of the agents we report results by averaging by 100 conversations, while for one of the agents we report only 25 conversations due to access restrictions.

5.2 Simulated Users

To instantiate a simulated user, we discuss how to initialize the preference model and train the interaction model; cf. Fig. 1.

To initialize the *preference model* we utilize the MovieLens dataset [13]. A user u in MovieLens reviews and rates movies from 0.5 to 5; we write M_u to denote the set of movies reviewed. Items rated at least 4 are regarded as liked, items rated 2 or below are regarded as disliked, and the remaining ratings in between are treated as neutral. For any simulated user, we construct the preference model by randomly sampling historical preferences of a real user from this dataset. Specifically, we sample 8 rated movies as items (of which at least one must be a liked item), and infer a personal knowledge graph (i.e., movie and genre preferences) from these items.

The *interaction model* is trained based on behaviors of real human users. Specifically, we collect conversations between the three conversational agents and human users using a crowdsourcing platform (Amazon Mechanical Turk). All conversational agents were deployed as a Telegram application.[7] Crowd workers were instructed to find the respective channel, engage in a conversation with the agent, and keep interacting with the agent until they receive a movie recommendation they like. For each conversational agent, we collected 25 successful dialogs (each with a different user) and paid $1.5 for each conversation. We regard a conversation successful if it covers the properties of both query formulation and set retrieval (mixed initiative is optional). Then, the utterances in each conversation are annotated manually with the actions listed in Table 1. For example, the system utterance "Could you give me one movie you like?" is labeled as *Elicit*. The annotations were performed by the paper's authors and disagreements were resolved through discussion. The conditional probabilities in Sect. 4.2 are estimated based on these empirical distributions.

6 EXPERIMENTAL EVALUATION

The main research question we seek to answer with our experiments is the following: *Can simulation be used to predict the performance of a conversational recommender agent with real users?* Each of the following subsections addresses a more specific sub-question.

[2] http://www.andchill.io/
[3] https://github.com/Sundar0989/Movie_Bot
[4] https://www.kaggle.com/rounakbanik/movie-recommender-systems/data
[5] https://www.ibm.com/cloud/watson-assistant/
[6] https://github.com/uber-research/plato-research-dialogue-system
[7] https://telegram.org/

Table 3: Simulation approaches used in our experiments. All use the same state transition modeling, NLU, and NLG.

Method	Interaction Model	Preference Model
QRFA-Single	QRFA (§4.2.2)	Single item (§4.3.1)
CIR6-Single	CIR6 (§4.2.3)	Single item (§4.3.1)
CIR6-PKG	CIR6 (§4.2.3)	PKG (§4.3.2)

Table 3 summarizes the three simulation approaches, which will be compared against real users, using both automatic and manual evaluation methods. Note that CIR6-PKG forces to empty the agenda once the user finds an item that they would like. This can significantly reduce the number of turns taken with the agent.

6.1 Characteristics of Conversations

(RQ1) How well do our simulation techniques capture the characteristics of conversations? To answer this question, we consider three statistical measures from the literature: (1) **AvgTurns**: the average number of dialogue turns [21]; (2) **UserActRatio**: the ratio of user and system acts [29], which is a measure of user participation; (3) **DS-KL**: a dissimilarity metric based on Kullback-Leibler divergence, which can be regarded as a measure of dialogue style [25]:

$$DS(P||Q) = \frac{D_{KL}(P||Q) + D_{KL}(Q||P)}{2},$$

where P and Q are probability distributions over the different user actions observed in simulated and real dialogs, respectively, and D_{KL} is the KL-divergence between two distributions. Note that this is an unbounded metric; the closer the number to zero, the more similar two distributions are.

Table 4 presents the results. We find that the first two simulation approaches, QRFA-Single and CIR6-Single, resemble quite closely the characteristics of conversations with real users, in terms of all three metrics. As expected, CIR6-PKG tends to have significantly shorter average conversation length, since it terminates the dialog as soon as the user finds a recommendation they like. Because of this, the distribution of the actions is also reshaped, as witnessed by higher DS-KL scores. Interestingly, this method is the closest to real humans in terms of user participation (UserActRatio).

6.2 Performance Prediction

(RQ2) How well do the relative ordering of systems according to some measure correlate when using real vs. simulated users? To answer this question, we perform end-to-end evaluation using two automatic evaluation metrics that have been used in the literature to evaluate the performance of task-oriented conversational agents.

Reward: Motivated by ABUS [30], the reward function assigns *Full* points (20) for successful task completion and *Cost* (1) for every user turn. We equally assign points to functions of query formulations *Disclose* and *Refinement*, set retrieval actions *Inquire* and *Navigation*, and mixed-initiative (4 points for each). For agents not supporting any of these functions, we deduct the corresponding points from *Full*. For example, if an agent does not support navigate in set retrieval, *Full* is set to 16 points. We deem two consecutive *Repeat* actions as one turn given that some bots do not support multi-turn *Navigation*. In the end, the reward function is: Reward = $\max\{0, Full - Cost \cdot T\}$, where T is the number of user turns.

Success Rate: We measure success rate on the turn level [24] based on the appropriateness of agent actions. I.e., if the agent returns a wrong action, we deem this turn as a failure. Table 5 presents the results. To answer RQ2, we look at the orderings produced by each evaluation metric. In terms of Reward, all simulation approaches produce the same ordering, which is the same as the one obtained with real users. The absolute values are also quite close to real users, with the exception of CIR6-PKG. Since that approach terminates conversations earlier, it in a way models a more effective user behavior, which yields higher scores. In terms of Success Rate, all but one simulator, CIR6-PKG, agree with the agent ranking produced by real users. It should be noted that agents A and B are very close in terms of absolute scores, and CIR6-PKG in fact comes closest to real humans in terms of absolute numbers. However, this method flips the order of agents A and B. Let us point out that this is not unreasonable as A consistently ranked higher than B in terms of Reward.

It should be noted that these findings are based only on three systems. Nevertheless, even with that, we can make some interesting observations that underline the need for further research on automatic evaluation measures. Looking at the relative orderings of agents across the two column in Table 5, it is clear that the two metrics disagree. This is not a problem in itself, as they evaluate different aspects of conversational agents. It, however, remains an open question which single metric aligns best with user satisfaction.

6.3 Realisticity

(RQ3) Do more sophisticated simulation approaches (i.e., more advanced interaction and preference modeling) lead to more realistic simulation? Our working definition for a *realistic* simulation is to be indistinguishable from conversations performed by real users.

Specifically, we follow the multi-turn protocols defined in [18] to compare a pair of dialogues conducted with a given conversational agent. One of these dialogues is performed by a real user and the other is a simulated user. Using crowdsourcing, we compare a sample of 25 simulated dialogues for each method and agent pair ($25 \times 3 \times 3 = 225$ dialogues in total). Each of the sampled dialogues is coupled with a human dialog with the corresponding agent and is shown side-by-side (in random order) to three workers on Amazon MTurk. Workers are then asked to choose which of the two dialogs was performed by a human. Ties are permitted when annotators find it difficult to distinguish. Additionally, workers are requested to give a brief explanation behind their choice. Options without explanations are filtered out. We present the results in Table 6.

To answer our research question, first we look at the effects of more advanced interaction modeling (QRFA-Single vs. CIR6-Single). We find that our interaction model (CIR6) leads to substantially more wins (+6% overall) over the existing model (QRFA). Introducing personal knowledge graphs for preference modeling (CIR6-Single vs. CIR6-PKG) brings in further improvements (+3% overall) in terms of wins. We note that this is the best overall setting, even though not all agents benefit from this (specifically, agent **A**).

The results obtained using our best model (CIR6-PKG) are in fact quite remarkable, considering that 36% of human evaluators have mistaken it for a real user, and 23% of them could not decide whether it was a real user or not.

Table 4: Comparison of the characteristics of dialogs with real and simulated users, for different conversational agents (A–C).

Method	AvgTurns			UserActRatio			DS-KL		
	A	B	C	A	B	C	A	B	C
Real users	9.20	14.84	20.24	0.374	0.501	0.500	-	-	-
QRFA-Single	10.52	12.28	17.51	0.359	0.500	0.500	0.027	0.056	0.029
CIR6-Single	9.44	12.75	15.92	0.382	0.500	0.500	0.055	0.040	0.025
CIR6-PKG	6.16	9.87	10.56	0.371	0.500	0.500	0.075	0.056	0.095

Table 5: Performance of conversational agents using real vs. simulated users, in terms of Reward and Success Rate. We show the relative ordering of agents (A–C), with evaluation scores in parentheses.

Method	Reward	Success Rate
Real users	**A** (8.88) > **B** (7.56) > **C** (6.04)	**B** (0.864) > **A** (0.833) > **C** (0.727)
QRFA-Single	**A** (8.04) > **B** (7.41) > **C** (6.30)	**B** (0.836) > **A** (0.774) > **C** (0.718)
CIR6-Single	**A** (8.64) > **B** (8.28) > **C** (6.01)	**B** (0.822) > **A** (0.807) > **C** (0.712)
CIR6-PKG	**A** (11.12) > **B** (10.65) > **C** (9.31)	**A** (0.870) > **B** (0.847) > **C** (0.784)

Table 6: Side-by-side comparison results, with human evaluators guessing which of two dialogs with a given conversational agent (A–C) was performed by a simulated user (Win) vs. a real one (Loss); a Tie is given when the evaluator could not decide.

	A			B			C			All		
	Win	Lose	Tie	Win	Lose	Tie	Win	Lose	Tie	Win	Lose	Tie
QRFA-Single	20	39	16	22	33	20	19	43	13	61 (27%)	115 (51%)	49 (22%)
CIR6-Single	27	30	18	23	33	19	26	40	9	76 (33%)	103 (46%)	46 (21%)
CIR6-PKG	22	39	14	27	29	19	32	25	18	81 (36%)	93 (41%)	51 (23%)

Table 7: Classification of comments accompanying decisions when choosing which of two conversations was made by a real user versus a simulated one in a side-by-side comparison. The columns are (Rea)listicity, (Eng)agement, (Emo)tion, (Res)ponse, (Gra)mmar, and (Len)gth.

	Style			Content		
	Rea.	Eng.	Emo.	Res.	Gra.	Len.
QRFA-Single	77	38	8	39	10	10
CIR6-Single	76	31	8	53	15	3
CIR6-PKG	74	33	15	34	14	9
All	227	102	31	126	39	22

7 FURTHER ANALYSIS

Recall that in our last experiment crowd workers were tasked with deciding, in a side-by-side experiment, which of two dialogs were performed by a real user vs. a simulated one (a "bot"). In addition to making a simple choice, they were also asked to briefly explain their reasoning. In this section, we further analyze these comments, in order to gain a better understanding of what traits of human behavior could be incorporated in a simulator in the future.

Based on an initial analysis of the comments, we came up with a coding scheme, which distinguishes between two main categories, dialogue Style and Content. We further subdivide Style into the following three categories: (1) **Realisticity** is associated with how realistic or human-sounding a dialog is. For example, "*User 2 seems a bit more human-like and realistic*" and "*The user is more genuine and stubborn about his requests which seem very natural.*" (2) **Engagement** is about the involvement of the user in the conversation, e.g., "*There appears to be more attempts at dialogue in the second conversation that seems more human*" and "*The first one is authentic and adds their opinion to each statement.*" (3) **Emotion** refers to expressions of feelings or emotions. For example, "*The user in dialogue 1 shows emotions like when he shows how he loves mila*" and "*Dialogue 2 expresses feeling, robots have no feelings.*"

We also distinguish between three Content subcategories: (1) **Response** refers to cases where the user does not seem to understand the agent correctly ("*some badly answered answers*") or repetitively asks the same question ("*the first one was too repetitive*"). (2) **Grammar** is about language usage, including spelling and punctuation. For example, "*User 1 made a spelling mistake with anna karinina, which i doubt a bot would do,*" and "*The user makes a typo in the left dialog.*" (3) **Length** concerns the length of reply ("*Very short and simple to the point*") or of a conversation ("*they had a longer conversation*").

Table 7 presents the statistics. Note that the numbers here do not mean success or failure; they merely indicate how often each aspect was considered when deciding whether the user in the conversation was a human or a bot. We hypothesize that the biggest gains in creating more human-like simulations lie in improving the aspects that were mentioned most. These are, in order: Realisticity (41%), Response (23%), and Engagement (19%), where the percentages are calculated with respect to all annotations. To improve Realisticity, one could imagine using a more natural tone for expressing preferences. The Response aspect may be enhanced by keeping a better track of conversation history and by generating more varied responses. As for Engagement, we observed that humans tend to continue the discussion and explore the space of options, even after they have found a recommendation they liked.

8 CONCLUSIONS AND FUTURE DIRECTIONS

We have introduced a simulation framework that enables large scale automatic evaluation of conversational recommender systems. Our simulation approaches, equipped with a preference model, interaction model, NLG, and NLU, are capable of generating human-like responses. We evaluate them by comparing three existing conversational movie recommender systems. The results indicate that the preference model and task-specific interaction models can achieve high correlation between automatic and human evaluations.

This work represents a first important step towards evaluating conversational information access systems using simulation. We see a number of directions for extending it in future work. First, we wish to generalize our findings by conducting simulations in multiple domains. Even though nothing is domain specific in our approach, the limited availability of conversational services in other domains represents a challenge. Second, there is a lack of automatic evaluation metrics for conversational information access. Our results indicate that it is possible to obtain the same relative ranking of agents via simulation than with real users for a given metric. However, current evaluation metrics do not agree with each other on how to rank agents. Thus, an important future research objective is to develop evaluation metrics that better align with user expectations and satisfaction. Additionally, we see potential in using simulation to debug conversational agents. For example, one could identify the type of actions where the system fails or locate points in a sequence of dialog turns when a given performance metric tends to drop. Then, by providing a sequence of utterances up to a given point, possible continuations of the dialog could be evaluated from that point. Third, some of our components rely on hand-crafted heuristics or make simplifying assumptions. For example, our NLG generates natural language responses with a template-based model. In the future, we wish to improve each individual component by investigating advanced natural language processing techniques. For example, for NLG we could using deep learning methods to learn to generate more human-like responses. Finally, motivated by the insights from our side-by-side evaluation, we wish to equip our simulated users with more personalized traits, such as emotion, engagement, and patience.

REFERENCES

[1] Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. 2019. Asking Clarifying Questions in Open-Domain Information-Seeking Conversations. In *Proc. of SIGIR '19*. 475–484.

[2] Layla El Asri, Jing He, and Kaheer Suleman. 2016. A Sequence-to-Sequence Model for User Simulation in Spoken Dialogue Systems. In *Proc. of Interspeech '16, 2016*. 1151–1155.

[3] Leif Azzopardi, Mateusz Dubiel, Martin Halvey, and Jeffery Dalton. 2018. Conceptualizing Agent-human Interactions during the Conversational Search Process. In *Proc. of CAIR '18*.

[4] Krisztian Balog and Tom Kenter. 2019. Personal Knowledge Graphs: A Research Agenda. In *Proc. of ICTIR '19*. 217–220.

[5] Anja Belz and Ehud Reiter. 2006. Comparing Automatic and Human Evaluation of NLG Systems. In *Proc. of EACL '06*.

[6] Keping Bi, Qingyao Ai, Yongfeng Zhang, and W Bruce Croft. 2019. Conversational Product Search Based on Negative Feedback. In *Proc. of CIKM '19*. 359–368.

[7] Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A Survey on Dialogue Systems: Recent Advances and New Frontiers. *SIGKDD Explor. Newsl.* 19, 2 (Nov. 2017), 25–35.

[8] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards Conversational Recommender Systems. In *Proc. of KDD '16*. 815–824.

[9] Grace Chung. 2004. Developing a Flexible Spoken Dialog System Using Simulation. In *Proc. of ACL '04*.

[10] J. Shane Culpepper, Fernando Diaz, and Mark D. Smucker. 2018. Research Frontiers in Information Retrieval: Report from the Third Strategic Workshop on Information Retrieval in Lorne (SWIRL 2018). *SIGIR Forum* 52, 1 (2018), 34–90.

[11] Jeff Dalton, Chenyan Xiong, and Jamie Callan. 2019. TREC Conversational Assistance Track. http://www.treccast.ai/.

[12] David Griol, Javier Carbó, and José M. Molina. 2013. An Automatic Dialog Simulation Technique to Develop and Evaluate Interactive Conversational Agents. *Appl. Artif. Intell.* 27, 9 (oct 2013), 759–780.

[13] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4, Article 19 (2015).

[14] Katja Hofmann, Lihong Li, and Filip Radlinski. 2016. Online Evaluation for Information Retrieval. *Found. Trends Inf. Retr.* 10, 1 (2016), 1–117.

[15] Dan Jurafsky and James H. Martin. 2019. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, 3nd Edition draft.* Prentice Hall, Pearson Education International.

[16] Diane Kelly. 2009. Methods for Evaluating Interactive Information Retrieval Systems with Users. *Found. Trends Inf. Retr.* 3, 1–2 (jan 2009), 1–224.

[17] Florian Kreyssig, Iñigo Casanueva, Pawel Budzianowski, and Milica Gasic. 2018. Neural User Simulation for Corpus-based Policy Optimisation of Spoken Dialogue Systems. In *Proc. of SIGDIAL '18*. 60–69.

[18] Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016. Deep Reinforcement Learning for Dialogue Generation. In *Proc. of EMNLP '16*. 1192–1202.

[19] Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial Learning for Neural Dialogue Generation. In *Proc. of EMNLP '17*. 2157–2169.

[20] Xiujun Li, Yun-Nung Chen, Lihong Li, Jianfeng Gao, and Asli Celikyilmaz. 2017. End-to-End Task-Completion Neural Dialogue Systems. In *Proc. of IJCNLP '17*. 733–743.

[21] Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation. In *Proc. of EMNLP '16*. 2122–2132.

[22] David Martin Maxwell. 2019. *Modelling search and stopping in interactive information retrieval.* Ph.D. Dissertation. University of Glasgow.

[23] Alexandros Papangelis, Yi-Chia Wang, Piero Molino, and Gokhan Tur. 2019. Collaborative Multi-Agent Dialogue Model Training Via Reinforcement Learning. In *Proc. of SIGDIAL '19*. 92–102.

[24] Baolin Peng, Xiujun Li, Jianfeng Gao, Jingjing Liu, and Kam-Fai Wong. 2018. Deep Dyna-Q: Integrating Planning for Task-Completion Dialogue Policy Learning. In *Proc. of ACL '18*. 2182–2192.

[25] Olivier Pietquin and Helen Hastie. 2013. A survey on metrics for the evaluation of user simulations. *The Knowledge Engineering Review* 28, 1 (2013), 59âĂŞ73.

[26] Chen Qu, Liu Yang, W. Bruce Croft, Yongfeng Zhang, Johanne R. Trippas, and Minghui Qiu. 2019. User Intent Prediction in Information-Seeking Conversations. In *Proc. of CHIIR '19*. 25–33.

[27] Filip Radlinski and Nick Craswell. 2017. A Theoretical Framework for Conversational Search. In *Proc. of CHIIR '17*. 117–126.

[28] Mark Sanderson. 2010. Test Collection Based Evaluation of Information Retrieval Systems. *Found. Trends Inf. Retr.* 4, 4 (2010), 247–375.

[29] Jost Schatzmann, Kallirroi Georgila, and Steve Young. 2005. Quantitative Evaluation of User Simulation Techniques for Spoken Dialogue Systems. In *Proc. of SIGDIAL '05*. 45–54.

[30] Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. 2007. Agenda-based User Simulation for Bootstrapping a POMDP Dialogue System. In *Proc. of NAACL-Short '07*. 149–152.

[31] Jost Schatzmann, Karl Weilhammer, Matt Stuttle, and Steve Young. 2006. A Survey of Statistical User Simulation Techniques for Reinforcement-Learning of Dialogue Management Strategies. *Knowl. Eng. Rev.* 21, 2 (June 2006), 97–126.

[32] Iulian Vlad Serban, Ryan Lowe, Peter Henderson, Laurent Charlin, and Joelle Pineau. 2018. A Survey of Available Corpora For Building Data-Driven Dialogue Systems: The Journal Version. *D & D* 9, 1 (2018), 1–49.

[33] Johanne R. Trippas. 2019. *Spoken Conversational Search: Audio-only Interactive Information Retrieval.* Ph.D. Dissertation. RMIT University.

[34] Svitlana Vakulenko. 2019. *Knowledge-based Conversational Search.* Ph.D. Dissertation. TU Wien.

[35] Svitlana Vakulenko, Kate Revoredo, Claudio Di Ciccio, and Maarten de Rijke. 2019. QRFA: A Data-Driven Model of Information Seeking Dialogues. In *Advances in Information Retrieval*. 541–557.

[36] Liu Yang, Minghui Qiu, Chen Qu, Jiafeng Guo, Yongfeng Zhang, W. Bruce Croft, Jun Huang, and Haiqing Chen. 2018. Response Ranking with Deep Matching Networks and External Knowledge in Information-Seeking Conversation Systems. In *Proc. of SIGIR '18*. 245–254.

[37] Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W. Bruce Croft. 2018. Towards Conversational Search and Recommendation: System Ask, User Respond. In *Proc. of CIKM '18*. 177–186.