Advances and Challenges in Conversational Recommender Systems: A Survey

Chongming Gao[a], Wenqiang Lei[b,*], Xiangnan He[a], Maarten de Rijke[c,d] and Tat-Seng Chua[b]

[a] *University of Science and Technology of China*
[b] *National University of Singapore*
[c] *University of Amsterdam, Amsterdam, The Netherlands*
[d] *Ahold Delhaize, Zaandam, The Netherlands*

ARTICLE INFO

Keywords:
conversational recommendation system
interactive recommendation
preference elicitation
multi-turn conversation strategy
exploration-exploitation

ABSTRACT

Recommender systems exploit interaction history to estimate user preference, having been heavily used in a wide range of industry applications. However, static recommendation models are difficult to answer two important questions well due to inherent shortcomings: (a) What exactly does a user like? (b) Why does a user like an item? The shortcomings are due to the way that static models learn user preference, i.e., without explicit instructions and active feedback from users. The recent rise of conversational recommender systems (CRSs) changes this situation fundamentally. In a CRS, users and the system can dynamically communicate through natural language interactions, which provide unprecedented opportunities to explicitly obtain the exact preference of users.

Considerable efforts, spread across disparate settings and applications, have been put into developing CRSs. Existing models, technologies, and evaluation methods for CRSs are far from mature. In this paper, we provide a systematic review of the techniques used in current CRSs. We summarize the key challenges of developing CRSs in five directions: (1) Question-based user preference elicitation. (2) Multi-turn conversational recommendation strategies. (3) Dialogue understanding and generation. (4) Exploitation-exploration trade-offs. (5) Evaluation and user simulation. These research directions involve multiple research fields like information retrieval (IR), natural language processing (NLP), and human-computer interaction (HCI). Based on these research directions, we discuss some future challenges and opportunities. We provide a road map for researchers from multiple communities to get started in this area. We hope this survey can help to identify and address challenges in CRSs and inspire future research.

1. Introduction

Recommender systems have become an indispensable tool for information seeking. Companies such as Amazon and Alibaba, in e-commerce, Facebook and Wechat, in social networking, Instagram and Pinterest, in content sharing, and YouTube and Netflix, in multimedia services, all have the need to properly link items (e.g., products, posts, and movies) to users. An effective recommender system that is both accurate and timely can help users find the desired information and bring significant value to the business. Therefore, the development of recommendation techniques continues to attract academic and industrial attention.

Traditional recommender systems, which we call *static recommendation models* in this survey, primarily predict a user's preference towards an item by analyzing past behaviors offline, e.g., click history, visit log, ratings on items. Early methods, such as collaborative filtering (CF) [168, 169], logistic regression (LR) [143], factorization machine (FM) [163], and gradient boosting decision tree (GBDT) [88], have been intensively used in practical applications due to the efficiency and interpretability. Recently, more complicated but powerful neural networks have been developed, including Wide & Deep [31], neural collaborative filtering (NCF) [67], deep interest network (DIN) [257], tree-based deep model (TDM) [266], and graph convolutional networks (GCNs) [235, 218, 66].

Inherent Disadvantages of Static Recommendations. Static recommendation models are typically trained offline on historical behavior data, which are then used to serve users online [36]. Despite their wide usage, they fail to answer two important questions:

1. *What exactly does a user like?* The learning process of static models is usually conducted on historical data, which may be sparse and noisy. Moreover, a basic assumption of static models is that all historical interactions represent user preference. Such a paradigm raises critical issues. First, users might not like the items they chose, as they may make wrong decisions [211, 212]. Second, the preference of a user may drift over time, which means that a user's attitudes towards items may change, and capturing the drifted preference from past data is even harder [80]. In addition, for cold users who have few historical interactions, modeling their preferences from data is difficult [97]. Sometimes, even the users themselves are not sure of what they want before being informed of the available options [210]. In short, a static model can hardly capture the precise preference of a user.
2. *Why does a user like an item?* Figuring out why a user

*Corresponding author

chongming.gao@gmail.com (C. Gao); wenqianglei@gmail.com (W. Lei); xiangnanhe@gmail.com (X. He); m.derijke@uva.nl (M. de Rijke); chuats@comp.nus.edu.sg (T. Chua)

http://chongminggao.me (C. Gao)

ORCID(s): 0000-0002-5187-9196 (C. Gao); 0000-0002-1086-0202 (M. de Rijke)

likes an item is essential to improve recommender model mechanisms and thus increase their ability to capture user preference. There are many factors affecting a user's decisions in real life [126, 19, 54]. For example, a user might purchase a product because of curiosity or being influenced by others [237]. Or it may be the outcome of deliberate consideration. It is common that different users purchase the same product but their motivations are different. Thus, treating different users equally or treating different interactions by the same user equally, is not appropriate for a recommendation model. In reality, it is hard for a static model to disentangle different reasons behind a user's consumption behavior.

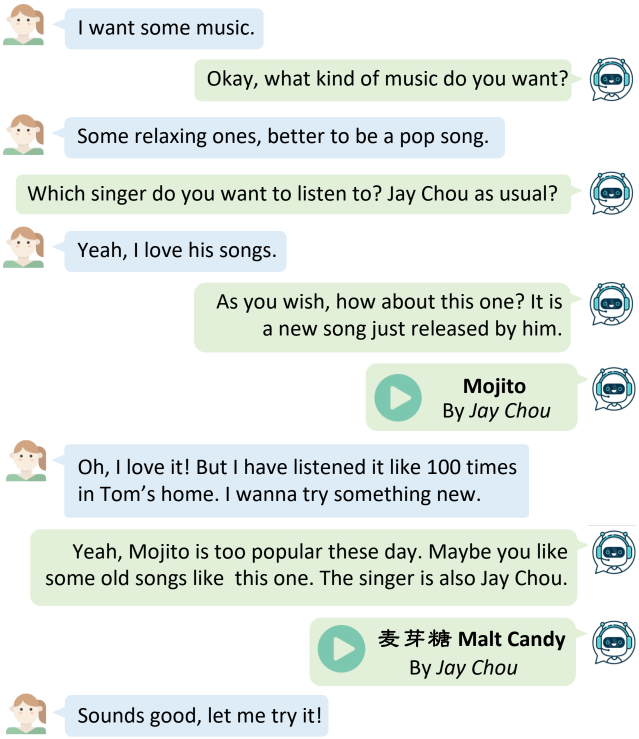
Even though much effort has been done to eliminate these problems, they make limited assumptions. For example, a common setting is to exploit a large amount of auxiliary data (e.g., social networks, knowledge graphs) to better interpret user intention [176]. However, these additional data may also be incomplete and noisy in real applications. We believe the key difficulty stems from the inherent mechanism: the static mode of interaction modeling fundamentally limits the way in which user intention can be expressed, causing an asymmetric information barrier between users and machines.

Introduction of CRSs. The emergence of conversational recommender systems (CRSs) changes this situation in profound ways. There is no widely accepted definition of CRS. In this paper, we define a CRS to be:

> *A recommendation system that can elicit the dynamic preferences of users and take actions based on their current needs through real-time multi-turn interactions.*

Our definition highlights a property of CRSs: *multi-turn interactions*. By a narrow definition, conversation means multi-turn dialogues in the form of written or spoken natural language; from a broader perspective, conversation means any form of interactions between users and systems, including written or spoken natural language, form fields, buttons, and even gestures [82]. Conversational interaction is a natural solution to the long-standing asymmetry problem in information seeking. Through interactions, CRSs can easily elicit the current preference of a user and understand the motivations behind a consumption behavior. Figure 1 shows an example of a CRS where a user resorts to the agent for music suggestions. Combining the user's previous preference (loving Jay Chou's songs) and the intention elicited through conversational interactions, the system can offer desired recommendations easily. Even if the produced recommendations do not satisfy the user, the system has chances to change recommendations based on user feedback.

Recently, attracted by the power of CRSs, many researchers have been on focusing on exploring this topic. These efforts are spread across a broad range of task formulation, in diverse settings and application scenarios. We collect the papers related to CRSs by searching for "Conversation* Recommend*" on DBLP[1] and visualize the statistics of them with regard to the published year and venue in Figure 2. There are 148 unique publications up to 2020, and we only visualize the top 10 venues, which contain 53 papers out of all 148 papers at all 89 venues. It is necessary to summarize these studies which put efforts into different aspects of CRSs.



I want some music.

Okay, what kind of music do you want?

Some relaxing ones, better to be a pop song.

Which singer do you want to listen to? Jay Chou as usual?

Yeah, I love his songs.

As you wish, how about this one? It is a new song just released by him.

Mojito
By *Jay Chou*

Oh, I love it! But I have listened it like 100 times in Tom's home. I wanna try something new.

Yeah, Mojito is too popular these day. Maybe you like some old songs like this one. The singer is also Jay Chou.

麦芽糖 Malt Candy
By *Jay Chou*

Sounds good, let me try it!

Figure 1: A toy example of a conversational recommender system in music recommendation.

Connections with Interactive Recommendations. Since the born of recommender systems, researchers have realized the importance of the human-machine interaction. Some studies propose interactive recommender systems [65, 205, 22, 264] and critiquing-based recommender systems [193, 195, 14, 179, 155, 26, 124, 123], which can be viewed as early forms of CRSs since they focus on improving the recommendation strategy online by leveraging real-time user feedback on previously recommended items.

In the setting of interactive recommendations, each recommendation is followed by a feedback signal indicating whether and how much the user likes this recommendation. However, interactive recommendations suffer from low efficiency, as there are too many items. An intuitive solution is to leverage attribute information of items, which is self-explanatory for understanding users' intention and can quickly narrow down candidate items. The critiquing-based recommender system is such a solution that is designed to elicit users' feedback on certain attributes, rather than items. Critiquing is like a salesperson who collects user preference by asking questions proactively on item attributes. For ex-

[1] https://dblp.org/search?q=conversation*%20recommend*

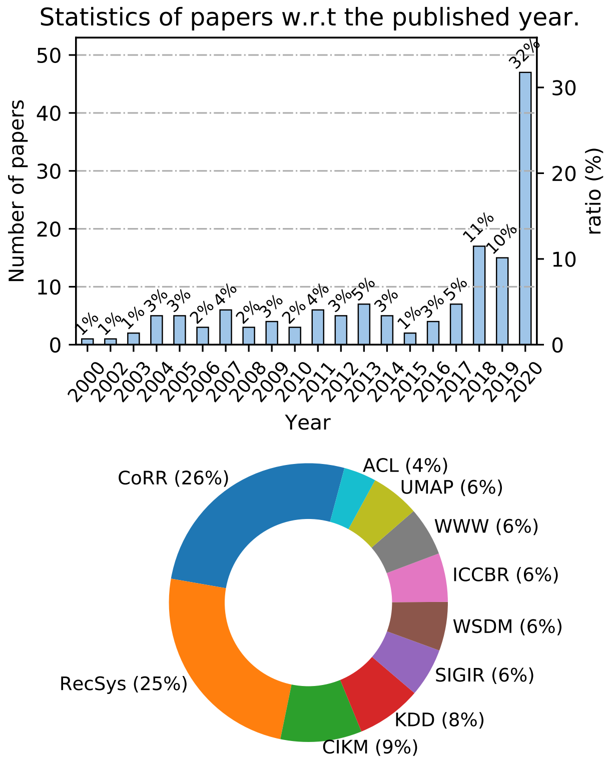


Figure 2: Statistics of the publications related to CRSs, grouped by the publication year and venue. Only the top 10 venues are used in the visualization.

ample, when seeking mobile phones, a user may follow the hint of the system and provides feedback such as "cheaper" or "longer battery life." Based on such feedback, the system will recommend more appropriate items; this procedure repeats several times until the user finds satisfactory items or gives up. The mechanism gives the system an improved ability to infer user preference and helps quickly narrow down recommendation candidates.

Though effective, existing interactive and critiquing methods have a limitation: the model makes a recommendation each time after receiving user feedback, which should be avoided as the recommendation should only be made when the confidence is high. This problem is solved in some CRSs by developing a conversation strategy determining when to ask and recommend [98, 100]. Besides, the interactive and critiquing methods are constrained by their representation ability since users can only interact with the system through a few predefined options. The integration of a conversational module in CRSs allows for more flexible forms of interaction, e.g., in the form of tags [34], template utterances [187], or free natural language [107]. Undoubtedly, user intention can be more naturally expressed and comprehended through a conversational module.

Connections with Other Conversational AI Systems. Besides CRSs, there are other conversational AI systems, e.g., task-oriented dialogue systems [23, 250, 151], social chatbots [128, 109, 223], conversational searching [200, 164, 161], and conversational question answering (QA) [265]. The common point of them is to utilize natural language as a powerful tool to convey information and thus to provide a natural user interface. Though these research topics all possess the keyword "conversation", the central tasks are different. For example, while task-oriented dialogue systems aim to fulfill a certain task in human-machine dialogue, the concentration of effort is mainly on handling information in the textural language-based dialogue, e.g., natural language understanding (NLU), dialogue state tracking (DST), dialogue policy learning (DPL), and natural language generation (NLG) [23, 250, 55]. In CRSs, however, the multi-turn conversation can be built on any form of interaction (e.g., form fields, buttons, and even gestures [82]) instead of merely textual form. Because CRSs concentrate on recommendation logic, the textual dialogue is just one possible means to convey information, i.e., it is auxiliary, not necessary. Although there are some CRSs implemented as end-to-end dialogue systems [107, 28], the human evaluation conducted by Jannach and Manzoor [81] suggests the performance is not ideal and more efforts should be put on improving both recommendation and language generation.

Other conversational AI systems can also be distinguished from CRSs by their specific scenarios. For instance, conversational searching focuses on analyzing the input query (in contrast to eliciting user preference in CRSs); conversational QA focuses on the single-turn question answering (in contrast to multi-turn interaction in CRSs). Therefore, it is essential to identify the central tasks and primary challenges in CRSs to help the beginner and future researchers set foot in this field and keep up with state-of-the-art technologies.

Focuses of This Survey. Although many studies have been done on CRSs, there is no uniform task formulation. In this survey, we present all CRSs as the general framework that consists of three decoupled components illustrated in Figure 3. Specifically, a CRS is made of a user interface, a conversation strategy module, and a recommendation engine. The user interface serves as a translator between the user and machine; generally, it extracts information from raw utterances of the user and transforms the information into machine-understandable representation, and it generates meaningful responses to the user based on the conversation strategy. The conversation strategy module is the brain of the CRS and coordinates the other two components; it decides the core logic of the CRS such as eliciting user preference, maintaining multi-turn conversations, and leading new topics. The recommendation engine is responsible for modeling relationships among entities (e.g., the user-item interaction or item-item linkage), learning and recording user preference on items and attributes of items, retrieving the required information.

There are many challenges in the three components, we summarize five main challenges as following.

- *Question-based User Preference Elicitation.* CRSs provide the opportunity to explicitly elicit user preference by asking questions. Two important questions are needed to be answered: (1) What to ask? (2) How to adjust the recommendations based on user response? The former fo-

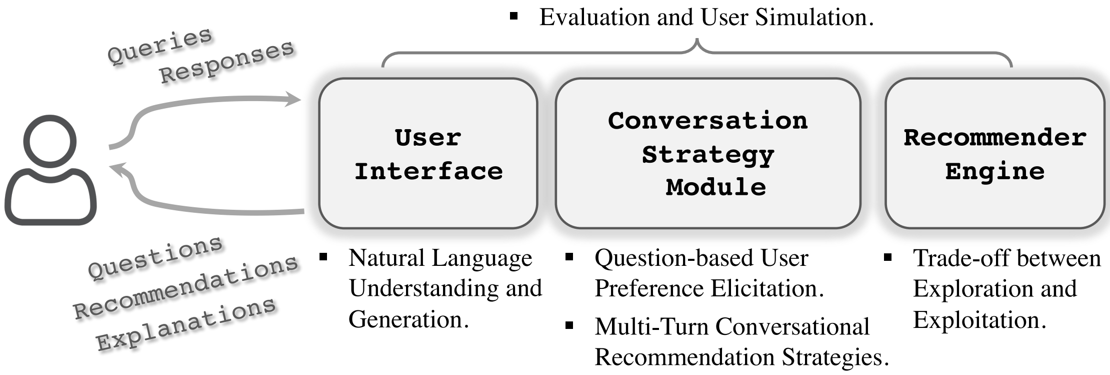

Evaluation and User Simulation.
Queries
Responses
Questions
Recommendations
Explanations
User Interface
Conversation Strategy Module
Recommender Engine
Natural Language Understanding and Generation.
Question-based User Preference Elicitation.
Multi-Turn Conversational Recommendation Strategies.
Trade-off between Exploration and Exploitation.

Figure 3: Illustration of the general framework of CRSs and our identified five primary challenges on the three main components.

cuses on constructing questions to elicit as much information as possible; the latter leverages the information in user response to make more appropriate recommendations.

- *Multi-turn Conversational Recommendation Strategies.* The system needs to repeatedly interact with a user and adapts to the user's response dynamically in multiple turns. An effective strategy concerns when to ask questions and when to make recommendations, i.e., let the model choose between (1) continuing to ask questions so as to further reduce preference uncertainty, and (2) generating a recommendation based on estimation of current user preference. Generally, the system should aim at a successful recommendation using the least number of turns, as users will lose their patience after too many turns [98]. Furthermore, some sophisticated conversational strategies try to proactively lead dialogues [222, 6], which can introduce diverse topics and tasks in CRSs [119, 263, 102, 215].
- *Natural Language Understanding and Generation.* Communicating like a human being continues to be one of the hardest challenges in CRSs. For understanding user interests and intentions, some CRS methods define the model input as pre-defined tags that capture semantic information and user preferences [34, 98, 100, 268]. Some methods extract the semantic information from users' raw utterances via slot filling techniques and represent user intents in slot-value pairs [245, 187, 162]. And for generating human-understandable responses, CRSs use many strategies such as directly providing a recommendation list [268, 245], incorporating recommended items in a rule-based natural language template [187, 98, 100]. Moreover, some researchers propose the end-to-end framework to enable CRSs to precisely understand users' sentiment and intentions from the raw natural language and to generate readable, fluent, consistent, and meaningful natural language responses [107, 119, 162, 28, 261].
- *Trade-offs between Exploration and Exploitation (E&E).* One problem of recommender systems is that each user can only interact with a few items out of the entire dataset. A large number of items that a user may be interested in will remain unseen by the user. For cold-start users (who have just joined the system and have zero or very few interactions), the problem is especially severe. Thanks to the interactive nature, CRSs can actively explore the unseen items to better capture the user preference. In this way, users can benefit from having chances to express their intentions and obtain better-personalized recommendations. However, the process of exploration comes at a price. As users only have limited time and energy to interact with the system, a failed exploration will waste time and lose the opportunity to make accurate recommendations. Moreover, exposing unrelated items hurts user preference, compared to exploiting the already captured preference by recommending the items of high confidence [172, 106, 59]. Therefore, pursuing E&E trade-offs is a critical issue in CRSs.
- *Evaluation and User Simulation.* Evaluation is an important topic. Unlike static recommender models that are optimized on offline data, CRSs emphasize the user experience during dynamic interactions. Hence, we should not only consider the turn-level evaluation for both recommendation and response generation but also pay attention to the conversation-level evaluation. Besides, evaluating CRSs requires a large number of online user interactions, which are expensive to obtain [106, 80, 73]. Practical solutions include: (1) leveraging the off-policy evaluation which assesses the target policy using the logged data under the behavior policy [59, 80], and (2) directly introducing user simulators to replace the true users in evaluation [241, 186].

The five challenges are allocated to the corresponding component as illustrated in Figure 3, where trading off the E&E balance is exclusive to the recommender engine; handling natural language understanding and generation is exclusive to the conversation module. The rest three challenges are related to both the components. We illustrate in Table 1 the solutions of some classic CRSs that focus on these directions. Limited by space, we only give part of the classic studies here. We will further discuss existing solutions in the following sections.

Differences with Existing Related Surveys. Recently, A number of related survey papers have been published. There

Table 1
Five primary challenges in CRSs and part of the classic methods that contribute to these challenges.

Primary Challenges in CRSs	Contributions of Existing Studies	Classic Publications
Question-based User Preference Elicitation	Asking about items	[253, 35, 238, 271, 130, 198, 120]
	Asking about attributes	[130, 229, 219, 34, 245, 187, 98, 100, 261]
Multi-turn Conversational Strategies	Explicit strategies	[187, 245, 98, 226]
	Leading diverse topics	[119, 263]
Language Understanding and Generation	End-to-end dialogue systems	[107, 28, 261, 225, 141]
Exploration and Exploitation Trade-offs	Leveraging multi-armed bandits	[35, 243, 108, 238]
Evaluation and User Simulation	Evaluation	[59, 73]
	User simulation	[241, 186]

are survey papers focusing on certain cutting-edge aspects in recommender systems, such as the bias issues and debiasing methods [24], explainability/interpretability [244], evaluation issues [177], and novel methods that leverage deep neural networks [242, 220, 221], knowledge graphs [61], or reinforcement learning [2] to improve the ability of recommendation systems. Also, there are survey papers that summarize new frontiers in conversational AI systems, such as the advanced methods [23, 250, 55] and the evaluation issues [18, 41] in dialogue systems. However, there is only one survey paper published in 2020 that focuses on CRSs [82].

Jannach et al. [82], for the first time, delved into different aspects of CRSs and made a comprehensive survey of CRSs. Specifically, they categorize existing CRSs in various dimensions, for instance, in terms of interaction modalities (e.g., buttons or written language), supported tasks (e.g., recommend or explain), or the knowledge CRSs use in the background (e.g., item-related information or dialogue corpora). Their survey provides a structured description of the CRS. Therefore, the audience, after reading this survey, can answer *what a CRS is*, for example, what the input/output or the functions of a CRS are. However, they may be still unsure about *what the key challenges are*, or *what to do next*. In our survey, we not only give the review of the current progress on CRSs including the existing assumptions and exploration but also refine the problems in state-of-the-art methods and summarize five challenges. We are trying to answer the three questions above, and we hope to provoke deeper thought and spark new ideas for the audience.

Survey Organization. The remainder of this paper is organized as follows. In next several sections, we discuss the main challenges in CRSs. Specifically, in Section 2, we illustrate how CRSs can elicit user preferences by asking informative questions. In Section 3, we describe the strategies in CRSs to interact with users in a multi-turn conversation. In Section 4, we point out the problems and provide solutions in dialogue understanding and generation for CRSs. In Section 5, we discuss how CRSs can balance the exploration-exploitation trade-off. In Section 6, we explore metrics and present techniques for evaluating CRSs. In Section 7, we envision some promising future research directions. And in Section 8, we conclude this survey.

2. Question-based User Preference Elicitation

A user looking for items with specific attributes may get assess to them by actively searching. For instance, a user may search “iphone12 red 256gb”, where the key phrases “red” and “256gb” are the attributes of the item iPhone12. In this scenario, users construct a query themselves, and the performance relies on both the search engine and the user’s expertise in constructing queries. Even though there are efforts on helping users complete queries by suggesting possible options based on what they entered [125, 7, 38, 15], users still need to figure out appropriate query candidates. Besides, searching in this way requires users to be familiar with each item they want, which is not true in practice. Recommender systems introduce users to the potential items that they may like. However, traditional recommender systems can only utilize the static historical records as the input, which results in the two main limitations mentioned in mysecintro.

Fortunately, CRSs can bridge the gap between the search engine and recommender system. Empowered by real-time interactions, CRSs can proactively consult users by asking questions. And with the feedback returned by users, CRSs can directly comprehend users’ needs and attitudes towards certain attributes, hence making proper recommendations. Even if users are not satisfied with the recommended items, a CRS has the opportunity to adjust its recommendations in the interaction process.

Question-driven methods focus on the problem of *what to ask* in conversations. Generally, there are two kinds of methods: (1) asking about items [253, 35, 174], or (2) asking about attributes/topics/categories of items [98, 100].

2.1. Asking about Items

Early studies directly ask users for opinions about an item itself [253, 208, 35, 271, 198]. Unlike traditional recommender systems which need to estimate user preferences in advance, CRSs can construct and modify the user profile during the interaction process.

In traditional recommender system models, the recommended items are produced in a relatively stable way from all candidates. In the CRS scenario, the recommended items should be updated after the system receives feedback from a user and it could be a complete change in order to adapt to the user's real-time preferences. Hence, instead of merely updating parameters of models online, some explicit rules or mechanisms are required. We introduce three methods that can elicit users' attitudes towards items and can quickly adjust recommendations. Most of these methods did not use natural language in their user interface, but it can easily integrate an natural language-based interface to make a CRS.

Choice-based Methods. The main idea of choice-based preference elicitation is to recurrently let users choose their preferred items or item sets from the current given options. The common strategies include (1) choosing an item from two given options [174], (2) selecting an item from a list of given items [84, 60, 165], and (3) choosing a set of items from two given lists [120]. After the user chooses preferred items, the methods change the recommendations according to the user's choice. For example, Loepp et al. [120] use the matrix factorization (MF) model [8] to initialize the embedding vectors of users and items, then select two sets of items from the item embedding space as candidate sets and let a user choose one of the two sets. It is important to ensure that the two candidate sets are as different or distinguishable as possible. To achieve this, the authors adopt a factor-wise MF algorithm [8], which factorizes the user-item interaction matrix and obtains the embedding vectors one by one in decreasing order of explained variance. Hence, the factors, i.e., different dimensions of embedding vectors, are ordered by distinctiveness. Then, the authors iteratively select two item sets with only a single factor value varying. For example, if two factors represent the degree of *Humor* and *Action* of movies, respectively, then the two candidate sets are one set of movies with a high degree of *Humor* and another with a low degree of *Humor*, while the degree of *Action* of the two sets is fixed to the average level. When a user chooses one item set, the user's preference embedding vector is set to the average of the embedding vectors of the chosen items. The choice becomes harder as the interaction process continues. Users can choose to ignore the question, which means the users cannot tell the difference between the two item sets or they do not care about it. Carenini et al. [16] further explore other strategies to select query items, e.g., selecting the most popular or the most diverse items in terms of users' history.

Bayesian Preference Elicitation. In addition, there are studies based on a probabilistic view of preference elicitation, which has been researched for a long time [20, 11, 198]. Basically, there is a utility function or a score function $u(\mathbf{x}_j, \mathbf{u}_i)$ representing user i's preference for item j. Usually, it can be written as a linear function as

$$u\left(\mathbf{x}_j, \mathbf{u}_i\right) = \mathbf{x}_j^T \mathbf{u}_i. \tag{1}$$

In a Bayesian setting, user i's preference is modeled by a probabilistic distribution instead of a deterministic vector, which means that the vector $\mathbf{u}_i$ is sampled from a prior user belief $P\left(\mathcal{U}^{(i)}\right)$. Therefore, the utility of an item j for a user i is computed as the expectation:

$$\mathbb{E}\left[u\left(\mathbf{x}_j, \mathbf{u}_i\right)\right] = \int_{\mathbf{u}_i \sim \mathcal{U}^{(i)}} P(\mathbf{u}_i) u\left(\mathbf{x}_j, \mathbf{u}_i\right) d\mathbf{u}_i. \tag{2}$$

The item with the maximum expected utility for user i is considered as the recommendation items:

$$\arg\max_j \mathbb{E}\left[u\left(\mathbf{x}_j, \mathbf{u}_i\right)\right]. \tag{3}$$

Based on the utility function, the system can select some items to query. And the user belief distribution can be updated based on users' feedback. Specifically, given a user response r_i to the question q, the posterior user belief $P(\mathbf{u}_i|q, r_i)$ can be written as:

$$P(\mathbf{u}_i|q, r_i) = \frac{P\left(r_i \mid q, \mathbf{u}_i\right) P(\mathbf{u}_i)}{\int_{\mathcal{U}^{(i)}} P\left(r_i \mid q, \mathbf{u}_i\right) P(\mathbf{u}_i) d\mathbf{u}_i}. \tag{4}$$

As for the query strategy, i.e., selecting which items to ask, there are different criteria. For example, Boutilier [11] propose a partially observed Markov decision process (POMDP) framework as the sequential query strategy. And Vendrov et al. [198] and Guo and Sanner [62] use the expected value of information (EVOI) paradigm as a relatively myopic strategy to select items to query. Furthermore, the query type can be classified into two different types: (1) a pairwise comparison query, in which the users are required to choose what they prefer more between two items or two item sets [35, 62, 174]; or (2) a slate query, where users need to choose from multiple given options [198].

Interactive Recommendation. Interactive recommendation models are mainly based on reinforcement learning. Some researchers adopt a multi-armed bandit (MAB) algorithm [253, 35, 208]. The advantage is two-fold. First, MAB algorithms are efficient and naturally support conversational scenarios. Second, MAB algorithms can exploit the items that users liked before and explore items that users may like but never tried before. There are also researchers formulate the interactive recommendation as a meta learning problem which can quickly adapt to new tasks [271, 97]. A task here is to make recommendations based on several conversation histories. Meta learning methods and MAB-based methods have the capability of balancing exploration and exploitation. We will describe it later in Section 5.

Recently, researchers incorporate deep reinforcement learning (DRL) models into interactive recommender systems [252, 22, 224, 254, 72, 269, 27, 75, 111, 150, 264, 270, 204]. Unlike MAB-based methods which usually assume the user preference is unchanged during the interaction, DRL-based methods can model a dynamic preference and long-term utility. For example, Mahmood and Ricci [129] introduce a model-based techniques and use the policy iteration algorithm [190] to acquire an adaptive strategy. Model-free frameworks such as deep Q-network (DQN) [252, 254, 269, 264] and deep deterministic policy gradient (DDPG) [72] are used in interactive recommendation scenarios. Most reinforcement learning (RL)-based methods often suffer from low efficiency issues and cannot handle cold-start users. Zhou et al. [264] propose to integrate a knowledge graph into the interactive recommendation to solve these problems.

For more works that leverage RL in interactive recommender systems, we refer the interested readers to the comprehensive survey conducted by Afsar et al. [2].

However, directly requiring items is inefficient for building the user profile because the candidate item set is large. In real-world CRS applications, users will get bored as the number of conversation turns increases. It is more practical to ask attribute-centric questions, i.e., to ask users whether they like an attribute (or topic/category in some works), and then make recommendations based on these attributes [245, 98]. Therefore, the estimation and utilization of a user's preferences towards attributes become a key research issue.

2.2. Asking about Attributes

Asking about attributes is more efficient because whether users like or dislike an attribute can significantly reduce the recommendation candidates. The challenge is to determine a sequence of attributes to ask so as to minimize the uncertainty of current user needs [138, 192]. The aforementioned critiquing-based methods fall into this category. Besides, there are other kinds of methods, we introduce some mainstream branches as below.

2.2.1. Fitting Patterns from Historical Interaction

A conversation can be deemed as a sequence of entities including consumed items and mentioned attributes, and the objective is to learn to predict the next attribute to ask or the next item to recommend. Therefore, the sequential neural network such as the gated recurrent unit (GRU) model [32] and the long short term memory (LSTM) model [69] can be naturally adopted in this setting, due to its ability to capture long and short term dependency in user behavioral patterns.

An exemplar work is the question & recommendation (Q&R) model proposed by Christakopoulou et al. [34], where the interaction between the system and a user is implemented as a selection system. In each turn, the system asks the user to choose one or more distinct topics (e.g., NBA, Comics, or Cooking) from the given list, and then recommends items in these topics to the user. It contains a trigger module to decide whether to ask a question about attributes or to make a recommendation. The triggering mechanism can be as simple as a random mechanism or can be more sophisticated, i.e., using criteria capturing the user's state, or even be user-initiated. At the t-th time step, the next topic q that user click can be predicted based on the user's watching history $e_1, \ldots, e_T$ as: $P\left(q \mid e_1, \ldots, e_T\right)$. After user clicking a topic q, the model can recommend an item r based on the conditional probability written as: $P\left(r \mid e_1, \ldots, e_T, q\right)$. Both of the two conditional probabilities are implemented as the GRU architecture [32]. This algorithm is deployed on YouTube, for obtaining preferences from cold-start users.

Zhang et al. [245] propose a "System Ask User Response" (SAUR) paradigm. For each item, they utilize the rich review information and convert a sentence containing an aspect-value pair to a latent vector via the GRU model. Then they adopt a memory module with attention mechanism [184, 93, 137] to perform both the next question generation task (determining which attribute to ask) and the next item recommendation task. Again, they also develop a heuristic trigger to decide whether it is the time to display the top-n recommended items to users or to keep asking questions about attributes. One limitation of the work is that the authors assume all information in reviews can support the purchasing behavior, however it is not true as users may complain certain aspects of the purchased items, e.g., a user may write "64 Gigabytes is not enough". Using information without discrimination will mislead the model and deteriorate the performance.

The utterances produced by the system, i.e., the questions, are constructed with predefined language patterns or templates, meaning that what the system needs to pay attention to are only the aspect and the value. This is a common setting in state-of-the-art CRS studies because the core task here is recommendation instead of language generation [34, 98, 100].

Note that these kinds of methods have a common disadvantage: learning from historical user behaviors cannot aid understanding the logic behind the interaction. As interactive systems, these models do not consider how to react to feedback when users reject the recommendation, i.e., they just try to fit the preferences in historical interaction and do not consider an explicit strategy to deal with different feedback.

2.2.2. Reducing Uncertainty

Unlike sequential neural network-based methods that do not have an explicit strategy to handle all kinds of user feedback, some studies try to build a straightforward logic to narrow down item candidates.

Critiquing-based Methods. The aforementioned critiquing model is typically equipped with a heuristic tactic to elicit user preference on attributes [26, 219, 124, 123]. In traditional critiquing models, where the critique on an attribute value (e.g., "not red" for color or "less expensive" for price)

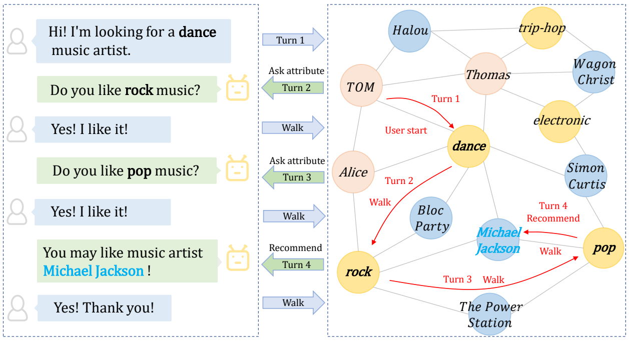


Figure 4: An illustration of interactive path reasoning in the conversational path reasoning (CPR) model. Credits: Lei et al. [100].

is used for reconstructing the candidate set by removing the items with unsatisfied attributes [26, 134, 180, 199, 14, 179]. The neural vector-based methods take the criticism into the latent vector, which is responsible for generating both the recommended items and the explained attributes. For example, Wu et al. [219] propose an explainable neural collaborative filtering (CE-NCF) model for critiquing. They use the NCF model [67] to encode the preference of a user i for an item j as a latent vector $\hat{\mathbf{z}}_{i,j}$, then $\hat{\mathbf{z}}_{i,j}$ is used for producing the rating score $\hat{r}_{i,j}$ as well as the explained attribute vector $\hat{\mathbf{s}}_{i,j}$. The attributes are composed of a set of key-phrases such as "golden, copper, orange, black, yellow," and each dimension of $\hat{\mathbf{s}}_{i,j}$ corresponds to a certain attribute. When a user dislikes an attribute and critique it in real-time feedback, the system updates the explained attribute vector $\hat{\mathbf{s}}_{i,j}$ by setting the corresponding dimension to zero. Then the updated vector $\tilde{\mathbf{s}}_{i,j}$ is used to update the latent vector $\hat{\mathbf{z}}_{i,j}$ to be $\tilde{\mathbf{z}}_{i,j}$. Consequently, the recommendation score is updated to be $\tilde{r}_{i,j}$. Following this setting, Luo et al. [124] change the base NCF model [67] to be a variational autoencoder (VAE) model, and this generative model can help the critiquing system have better computational efficiency, improved stability, and faster convergence.

Reinforcement Learning-driven Methods. Reinforcement learning is also used in CRSs to select the appropriate attributes to ask [187, 98, 100]. Empowered by a deep policy network, the system not only selects the attributes but also determine a controlling strategy on when to change the topic of the current conversation; we will elaborate this in Section 3.1 where we describe how reinforcement learning helps the system form a multi-turn conversational strategy.

Graph-constrained Candidates. Graph is a prevalent structure to represent relationship of different entities. It is natural to utilize graphs to sift items given a set of attributes. For example, Lei et al. [100] propose an interactive path reasoning algorithm on a heterogeneous graph on which users, items, and attributes are represented as nodes and an edge connected two nodes represented a relationship between two nodes, e.g., a user purchased an item, or an item has a certain value for an attribute. With the help of the graph, a conversation can be converted to a path on the graph, as illustrated in Figure 4. The authors compare the uncertainty of preference for attributes and choose the attributes with the maximum uncertainty to ask. Here, the preference for a certain attribute is modeled by the average preference for items that have this attribute. Hence, the searching space and overhead of the algorithm can be significantly reduced by utilizing the graph information. There are other studies that apply graph neural networks (GNNs) to learn a powerful representation of both items and attributes, so the semantic information in the learned embedding vectors can help end-to-end CRS models generate appropriate recommendations. For example, the GCN model and its variants [91, 171] are adopted on the knowledge graph in recent CRS models [28, 261, 225, 112].

Other Methods. There are other attempts to make recommendations based on user feedback on attributes. For example, Zou et al. [268] proposed a question-driven recommender system based on an extended matrix factorization model, which merely considers the user rating data, to combine real-time feedback from users.

The basic assumption is that if a user likes an item, then he/she will like the attributes of this item. Thereby, in each turn, the system will select the attribute that carries the maximum amount of uncertainty to ask. In other words, if an attribute is known to be shared by most items that a user likes, then it does not need to ask about this attribute. Similarly, there is no need to ask about the attributes that users dislike. Only if it is not sure whether a user likes an attribute, then asking about this attribute can provide the most amount of information. The parameters in matrices can be updated after users providing feedback. Besides, using ideas similar to aforementioned models based on asking items, MAB-based models [243, 108] and Bayesian approaches [130] are also developed in attribute-asking CRSs.

2.3. Section Summary

We list the common CRS models in Table 2, where the models are characterized by different dimensions, which are the asking entity (item or attribute), the asking mechanism, the type of user feedback, and the multi-turn strategy that we will describe in the next section.

In most interactive recommendations [270, 204, 240, 44] and critiquing methods [26, 219, 124, 123], the system keeps asking questions, and each question is followed by a recommendation. This process will only terminate when users quit with either being satisfied or impatient. The setting is unnatural and will likely hurt the user experience during the interaction process. Asking too many questions may let the interaction become an interrogation. Moreover, during the early stages of interaction, when the system has not confidently modeled the user preferences yet, recommendations with low confidence should not be exposed to the user [172]. In other words, there should be a multi-turn conversational strategy to control how to switch between asking and recommending, and this strategy should change dynamically in the interaction process.

Table 2
Characteristics of common CRS models in different dimensions. The strategy indicates whether the work considers an explicit strategy to control multi-turn conversations, e.g., whether to ask or recommend in the current turn.

Asking	Asking Mechanism	Basic Model	Type of User Feedback	Strategy	Publications
Items	Exploitation & Exploration	Multi-armed bandit	Rating on the given item(s)	No	[253, 35, 256, 213, 238]
	Exploitation & Exploration	Meta learning	Rating on the given item(s)	No	[271, 97]
	Maximal posterior user belief	Bayesian methods	Rating on the given item(s)	No	[198]
	Reducing uncertainty	Choice-based methods	Choosing an item or a set of items	No	[120, 84, 60, 165, 160]
Attributes	Exploitation & Exploration	Multi-armed bandit	Rating on the given attribute(s)	Yes	[243, 108]
	Reducing uncertainty	Bayesian approach	Providing preferred attribute values	No	[130, 229]
		Critiquing-based methods	Critiquing one/multiple attributes	No	[134, 180, 199, 14, 179] [155, 26, 219, 124, 123]
		Matrix factorization	Answering Yes/No for an attributes	No	[268]
	Fitting historical patterns	Sequential neural network	Providing preferred attribute values	Yes	[34, 245]
			Providing an utterance	No	[107, 28]
	Maximal reward	Reinforcement learning	Answering Yes/No for an attributes	Yes	[98, 100]
			Providing an utterance	Yes	[187, 194, 86]
				No	[162]
	Exploring graph-constrained candidates	Graph reasoning	Answering Yes/No for an attributes	Yes	[100]
			Providing an utterance	Yes	[28, 119]
				No	[261, 112]
			Providing preferred attribute values	Yes	[225]
				No	[141]

3. Multi-turn Conversational Strategies for CRSs

Question-driven methods focus on the problem of "*What to ask*", and the multi-turn conversational strategies discussed in this section focus on "*When to ask*" or a broader perspective, "*How to maintain the conversation*". A good strategy cannot only make the recommendation at the proper time (with high confidence) and adapt flexibly to users' feedback, but also maintain the conversation topics and adapt to different scenarios to make users feel comfortable in the interaction.

3.1. Conversation Strategies for Determining When to Ask and Recommend

Most CRS models do not carefully consider a strategy to determine whether to continue interrogating users by asking questions or to make a recommendation. However, a good strategy is essential in the interaction process so as to improve the user experience. The strategy can be a rule-based policy, i.e., making recommendations every k turns of asking questions [243], or a random policy [34], or a model-based policy [34].

In the SAUR model [245], a trigger is set to activate the recommendation module when the confidence is high. The trigger is simply implemented as a sigmoid function on the score of the most probable item, i.e., if the score of the candidate item is high enough, then the recommendation step is triggered, else the system will keep asking questions.

Though straightforward and easy to control, these strategies cannot capture rich semantic information, e.g., what topics are talking about now or how deep the topics have been explored. This information can directly affect the conversation topic. Thereby, a sophisticated strategy is necessary. Recently, reinforcement learning (RL) has been adopted by many interactive recommendation models for its potential of modeling the complex environment [252, 22, 224, 254, 269, 27, 75, 111, 150, 240, 264]. Therefore, it is natural to incorporate RL into the CRS framework [187, 98, 100, 194, 162, 86]. For instance, Sun and Zhang [187] propose a model called conversational recommender model (CRM) that uses the architecture of task-oriented dialogue system. In CRM, a belief tracker is used to track the users' input, and

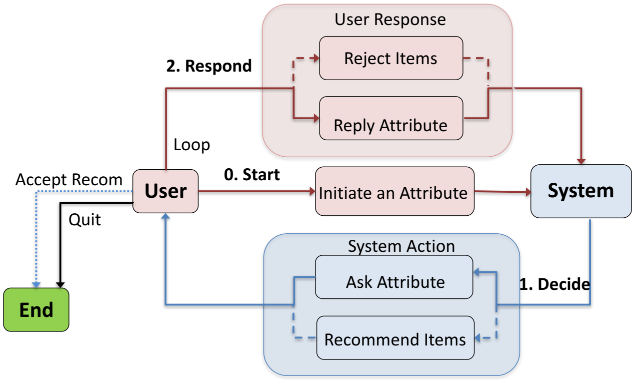


Figure 5: The estimation-action-reflection workflow. Credits: Lei et al. [98].

it outputs a latent vector representing the current state of the dialogue and the user preferences that have so far been captured. Afterward, the state vector of the belief tracker is input into a deep policy network to decide whether to recommend an item or to keep asking questions. Specifically, there are $l + 1$ actions: l actions for choosing one facet to ask and the last one is to yield a recommendation. The deep policy network uses the policy gradient method to make decisions. Finally, the model gets rewards from the environment, which includes user feedback towards the questions and the reward from the automatic evaluation of recommendation results.

However, the state modeled in CRM is a latent vector capturing the information of facet-values, which is hard to interpretable. In this respect, some studies explore better ways to construct the state of RL to make the multi-turn conversation strategy better adapt to an dynamic environment. For example, Lei et al. [98] propose an Estimation-Action-Reflection (EAR) framework, which assumes that the model should only ask questions at the right time. The right time, in their definition, is when (1) the item candidate space is small enough; (2) asking additional questions is determined to be less useful or helpful, from the perspective of either information gain or user patience; and (3) the recommendation engine is confident that the top recommendations will be accepted by the user. The workflow of the EAR framework is illustrated in Figure 5, where the system has to decide whether to continue to ask questions about attributes or to make a recommendation based on available information. To determine when to ask a question, they construct the state of the RL model to take into account four factors:

- Entropy information of each attribute among the attributes of the current candidate items. Asking attributes with a large entropy helps to reduce the candidate space, thus benefits finding desired items in fewer turns.
- User preference on each attribute. The attribute with a high predicted preference is likely to receive positive feedback, which also helps to reduce the candidate space.
- Historical user feedback. If the system has asked about a number of attributes for which the user gives approval, it may be a good time to recommend.
- Number of rest candidates. If the candidate list is short enough, the system should turn to recommend to avoid wasting more turns.

Building on these vectors capturing the current state, the RL model learns the proper timing to ask or recommend, which is more intelligent than a fixed heuristic strategy.

During the conversation, the recommendation module takes the items in the previous list of recommendations that are not chosen by users as the negative samples. However, Lei et al. [98] mention that this setting deteriorates the performance of the recommendation results. The reason, as they analyze it, is that rejecting the produced attribute does not mean that the user dislikes it: maybe the user does like it but overlooks it or just wants to try other new things.

Furthermore, Lei et al. [100] extend the EAR model by proposing the CPR model. By integrating the knowledge graph consisted of users, items, and attributes, they model conversational recommendation as an interactive path reasoning problem on the graph. A toy example of the generated conversation of the CPR model is shown in Figure 4. Unlike the EAR model where the attributes to be asked are selected irregular and unpredictable from all attribute candidates, CPR chooses attributes to be asked and items to be recommended strictly following the paths on the knowledge graph, which renders interpretable results.

In terms of the timing to ask or recommend, CRP makes an important improvement: the action space of the RL policy is only two — asking an attribute or making item recommendations. This largely reduces the difficulty of learning the RL policy. The CPR model is much more efficient than the EAR model due to the fact that the searching space of attributes in CPR is constrained by the graph. The integration of knowledge improves the multi-turn conversational reasoning ability.

3.2. Conversation Strategies from A Broader Perspective

Although learning from the query-answering interactions can enable the system to understand and respond to human query directly, the system still lacks intelligence. One reason is that most CRS models assume that users always bear in mind what they want, and the task is to obtain the preference through asking questions. However, users who resort to recommendation might not have a clear idea about what they really want. Just like a human asks a friend for suggestions on restaurants. Before that, he may not have a certain target in mind, and his decision can be affected by his friend's opinions. Therefore, CRSs should not only ask clarification questions and interrogate users, but also take responsibility for leading the topics and affecting users' mind. Towards this objective, some studies try to enrich CRSs certain personalities or endow CRSs the ability to lead the conversation, which can make the dialogues more attractive and more engaging. These efforts can also be found in the field of proactive conversation [140, 222, 6].

3.2.1. *Multi-topic Learning in Conversations*

Borrowing the idea from the proactive conversation, Liu et al. [119] present a new task which places conversational recommendation in the context of multi-type dialogues. In their model, the system can proactively and naturally lead a conversation from a non-recommendation dialog (e.g., question answering or chitchat) to a recommendation dialog, taking into account the user's interests and feedback. And during the interaction, the system can learn to flexibly switch between multiple goals. To address this task, they propose a multi-goal driven conversation generation (MGCG) framework, which consists of a goal planning module and a goal-guided responding module. The goal-planning module can conduct dialog management to control the dialog flow, which takes recommendation as the main goal and complete the natural topic transitions as the short-term goals. Specifically, given a user's historical utterances as context X and the last goal g_{t-1}, the module estimates the probability of changing the goal g_t of the current task as $P_{GC}(g_t \neq g_{t-1}|X, g_{t-1})$. The goal g_t of the current task is changed when the probability $P_{GC} > 0.5$ and remains to be g_{t-1} if $P_{GC} \leq 0.5$. Based on the current goal, the framework can produce responses from an end-to-end neural network.

Learning a multi-type conversational model requires a dataset that supports multi-type dialogues. Therefore, Liu et al. [119] create a dataset, denoted as DuRecDial, with various types of interaction. In DuRecDial, two human workers are asked to conduct the conversation based on a given profile, which contains the information of age, gender, occupation, preferred domains, and entities. The workers must produce utterances that are consistent with their given profiles, and they are encouraged to produce utterances with diverse goals, e.g., question answering, chitchat, or recommendation. Then these dialogue data are labeled with goals and goal descriptions by templates and human annotation.

Further, Zhou et al. [263] release a topic-guided conversational recommendation dataset. They collect the review data from Douban Movie [2], a movie review website, to construct the recommended movies, topic threads, user profiles, and utterances. And they associate each movie with the concepts in ConceptNet [181], a commonsense knowledge graph, for providing rich topic candidates. Then they use rules to generate multi-turn conversations with diverse topics based on the user profile and topic candidates. Based on the proposed dataset, a new task of topic-guided conversational recommendation is defined as follows: given the user profile P_u, user interaction sequence I_u, historical utterances $s_1, \ldots, s_{k-1}$, and corresponding topic sequence $\{t_1, \ldots, t_{k-1}\}$, the system should: (1) predict the next topic t_k, or (2) recommend the movie i_k, and finally (3) produce a proper response s_k about the topic and with persuasive reasons.

3.2.2. *Special Ability: Suggesting, Negotiating, and Persuading*

There are miscellaneous tasks beyond the preference elicitation and recommendation for an intelligent interactive system, which require the CRS to possess different abilities to react in different scenarios. This is a high-level and abstract requirement. A lot of effort have put into helping the machine improve the topic's guiding ability. For instance, in conversational search, where traditional work has mainly attempted to better understand a user's information needs by resolving ambiguity, Rosset et al. [164] propose to lead the conversation with questions that a user may want to ask in the next step. For example, if a user queried "Nissan GTR Price," then the system can provide question suggestions include those that help the user complete a task ("How much does it cost to lease a Nissan GT-R?"), weigh options ("What are the pros and cons of the Nissan GT-R?"), explore an interesting related topic ("Is the Nissan GT-R the ultimate streetcar?"), or learn more details ("How much does 2020 Nissan GTR cost?"). These question suggestions can lead the user to an immersive search experience with diverse and fruitful future outcomes.

In addition, Lewis et al. [102] propose a system that is capable of engaging in the negotiations with users. They define the problem as an allocation problem: there are some items that need to be allocated to two people, where each item has a different value to a different person and people do not know the value of others. Hence, the two people have to converse and negotiate with each other to reach an agreement about the division of these items. Instead of optimizing relevance-based likelihood, the model should pursue a maximal profit for both parties. The authors use RL to tackle this problem. And they interleave RL updates with supervised updates to avoid that the models diverges from human language.

Wang et al. [215] develop a model that tries to persuade users to take certain actions, which is very promising for conversational recommendation. They train the model, according to conversational contexts, to learn and predict the 10 persuasion strategies (e.g., logical appeal or emotion appeal) used in the corpus. And they analyze which strategies are better conditioned on the background (personality, morality, value systems, willingness) of the user being persuaded.

Though some of these efforts are applied to specific application scenarios in dialogue systems, these techniques can be adopted in the multi-turn strategy in CRSs and thus push the development of CRSs.

3.3. Section Summary

The multi-turn conversation strategies of CRSs discussed in this section are summarized in Table 3. The main focus of the conversation strategy is to determine when to elicit user preference by asking questions and when to make recommendations. As a recommendation should only be made when the system is confident, an adaptive strategy can be more promising compared to a static one. Besides this core function, we introduce some strategies from a broader perspective. These strategies can extend the capability of CRSs by means of leading multi-topic conversations [119, 263] or showing special ability such as suggesting [164], negotiating [102], and persuading [215].

[2] https://movie.douban.com/

Table 3
The commonly used multi-turn strategies in CRSs.

Main Mechanism	Asking Method	When to ask and recommend	Determining X and Y	Publications
Asking questions	Explicit	Asking 1 turn; recommending 1 turn	Fixed	[34, 238]
		Asking X turn(s); recommending 1 turn	Fixed	[268]
			Adaptive	[187]
		Asking X turn(s); recommending Y turn(s)	Adaptive	[245, 98, 100, 108, 226]
	Implicit	Contained in natural language	Adaptive	[107, 28, 261, 263]
Leading diverse topics or explore special abilities				[119, 263, 164, 102, 215]

4. Dialogue Understanding and Generation in CRSs

An important direction of CRSs is to converse with humans in natural languages, thus understanding human intentions and generating human-understandable responses are critical. However, most CRSs only extract key information from processed structural data and present the result via rule-based template responses [245, 268, 98, 100]. This not only requires lots of labor to construct the rule or template but also make the result rely on the preprocessing. It also hurt user experience as the constrained interaction is unnatural in real-world applications. Recently, we have witnessed the development of end-to-end learning frameworks in dialogue systems, which have been studying for years to automatically handle the semantic information in raw natural language [55, 99, 85]. We will introduce these natural language processing (NLP) technologies in dialogue systems and describe how they help CRSs understand user intention and sentiment and generate meaningful responses.

4.1. Dialogue Understanding

Understanding users' intention is the key requirement for the user interface of a CRS, as downstream tasks, e.g., recommendation, rely heavily on this information. However, most CRSs pay attention to the core recommendation logic and the multi-turn strategy, while they circumvent extracting user intention from raw utterances and requires the preprocessed input such as rating scores [253, 35, 271, 97], YES/NO answers [268, 98, 100], or another type of value or orientation [34, 245] towards the queried items or attributes. This is unnatural in real-life human conversation and imposes constraints on user expression. Thereby, it is necessary to develop methods to extract semantic information in users' raw language input, either in an explicit or implicit way.

We introduce how dialogue systems use NLP technologies to address this problem and give the examples of CRSs that use these technology to understand user intention.

4.1.1. Slot Filling

A common way used in dialogue systems to extract useful information is to predefine some aspects of interest and use a model to fill out the values of these aspects from users' input, a.k.a, slot filling [39, 40, 233, 136, 232, 149]. Sun and Zhang [187] first consider extracting the semantic information from the raw dialogue in CRSs. They propose a belief tracker to capture the facet-value pairs, e.g., (*color, red*), from user utterances. Specifically, given a user utterance e_t at time step t, the input to the belief tracker is the n-gram vector $\mathbf{z}_t$, which is written as $\mathbf{z}_t = \text{n-gram}(e_t)$, where the dimension of $\mathbf{z}_t$ is the corpus size. This means that only the positions corresponding to the words in utterance e_t are set to 1, other positions will be set to 0. Suppose there are K types of facet-value pairs, for a given facet $m \in \{1, 2, \ldots, K\}$, the user's sequential utterances $\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_t$ are encoded by a LSTM model [69] to learn the latent vector f_m for this facet m. The size of vector f_m is set to the number of values, e.g., the number of available colors. The vector f_m capturing the facet-value information will be used in the recommendation module and policy network later. Besides, Ren et al. [162], Tsumita and Takagi [194] also employ recurrent neural networks (RNN)-based methods to extract the facet-value information as input for in downstream tasks in their CRSs.

However, explicitly modeling semantic information as aspect-value pairs can be a limitation in some scenarios where it is difficult and also unnecessary to do that. Besides, aspect-value pairs cannot precisely express information such as user intent or sentiment. Therefore, some recent CRSs use end-to-end neural frameworks to implicit learning the representation of users' intentions and sentiment.

4.1.2. Intentions and Sentiment Learning

Neural networks are famous for extracting features automatically, so it can be used to extract users' intentions and sentiment in CRSs. An classic example in CRSs is the end-to-end framework that proposed by Li et al. [107], which takes the user's raw utterances as input and directly produces the responses in the interaction. They collect the REDIAL dataset [3] through the crowdsourcing platform Amazon Mechanical Turk (AMT) [4]. They pair up AMT workers and give each of them a role. The movie seeker has to explain what kind of movie he/she likes, and asks for movie sug-

[3] https://redialdata.github.io/website/
[4] https://www.mturk.com/

gestions. The recommender tries to understand the seeker's movie tastes and recommends movies. All exchanges of information and recommendations are made using natural language; every movie mention is tagged using the "@" symbol to let the machine know it is a named entity. In this way, the dialogues in the REDIAL data contain the required semantic information that can help the model learn to answer users with recommendations and reasonable explanations. In addition, three questions are asked to provide labels for supervised learning: (1) Whether the movie was mentioned by the seeker, or was a suggestion from the recommender ("suggested" label). (2) Whether the seeker has seen the movie ("seen" label): one of *Have seen it*, *Haven't seen it*, or *Didn't say*. (3) Whether the seeker liked the movie or the suggestion ("liked" label): one of *Liked*, *Didn't like*, *Didn't say*. The three labels are collected from both the seeker and the recommender.

In this way, although the facet-value constraints are removed, all kinds of information including mentioned items and attributes, user attitude, and user interest are preserved and labeled in the raw utterance. And the CRS model needs to directly learn users' sentiment (or preferences), and it will make recommendations and generate responses based on the learned sentiment. The deep neural network-based model consists of four parts: (1) A hierarchical recurrent encoder implemented as a bidirectional GRU [32] that transforms the raw utterances into a latent vector with the key semantic information remained. (2) At each time a movie entity is detected (with the "@" identifier convention), an RNN model is instantiated to classify the seeker's sentiment or opinion regarding that entity. (3) An autoencoder-based recommendation module that takes the sentiment prediction as input and produces an item recommendation. (4) A switching decoder generating the response and deciding whether the name of the recommended item is included in the response. The model generates a complete sentence that might contain a recommended item to answer each user's utterance.

Beside using the RNN-based neural networks, there are some CRSs that adopt the convolutional neural network (CNN) model [162, 119], which has been proven to be very effective for modeling the semantics from raw natural language [90]. However, deep neural networks are often criticized to be non-transparent and hard to interpretable [13]. It is not clear how the deep language models can help CRSs in understanding user needs.

In order to answer this question, Penha and Hauff [152] investigate the bidirectional encoder representations from transformers (BERT) [42], a powerful technology for NLP pretraining developed by Google, to analyze whether its parameters can capture and store semantic information about items such as books, movies, and music for CRSs. The semantic information includes two kinds of knowledge needed for conducting conversational search and recommendation, namely content-based and collaborative-based knowledge. Content-based knowledge is knowledge that requires the model to match the titles of items with their content information, such as textual descriptions and genres. In contrast, collaborative-based knowledge requires the model to match items with similar ones, according to community interactions such as ratings. The authors use the three probes on the BERT model (i.e., tasks to examine a trained model regarding certain properties) to achieve the goal. And the result shows that both collaborative-based and content-based knowledge can be learned and remembered. Therefore, the end-to-end language model has potential as part of CRS models to interact with humans directly in real-world applications with complex contexts.

4.2. Response Generation

A natural language-based response of a CRS should at least meet two levels of standards. The lower level standard requires the generated language to be proper and correct; the higher level standard requires the response contains meaningful and useful information about recommended results.

4.2.1. Generating Proper Utterances in Natural Language

Many CRSs use template-based methods to generate responses in conversations [187, 98, 100]. However, template-based methods suffer from producing repetitive and inflexible output, and it require intense manual work. Besides, template-based responses could make users uncomfortable and hurt user experience. Hence, it is important to automate the response generation in CRSs to produce proper and fluent responses. This is also the objective of dialogue systems, so we introduce two veins of technologies for producing responses in dialogue systems:

Retrieval-based Methods. The basic idea is to retrieve the appropriate response from a large collection of response candidate. This problem can be formulated as a matching problem between an input user query and the candidate responses. The most straightforward method is to measure the inner-product of the feature vectors representing a query and a response [223]. A key challenge is to learn a proper feature representation [223]. One strategy is to use neural networks to learn the representation vectors from user query and candidate response, respectively. Then, a matching function is used to combine the two representations and output a matching probability [70, 191, 159, 49, 203]. An alternative strategy, in contrast, is to combine the representation vectors of query and response first, and then a neural method is used on the combined representation pair to further learn the interaction [209, 202, 146, 122]. These two strategies have their own advantages: the former is more efficient and suitable for online serving, while the latter is better at efficacy since the matching information is sufficiently preserved and mined [223].

Generation-based Methods. Unlike retrieval-based methods, which select existing responses from a database of template response, generation-based methods directly produce a complete sentence from the model. The basic generation model is a recurrent sequence-to-sequence model, which sequentially feeds in each word in the query as input, and then generates the output word one by one [189]. Compared to retrieval-based methods, generation-based methods have some challenges. First, the generated answer is not guaranteed to be a well-formed natural language utterance [228]. Second, even though the generated response may be grammatically correct, we can still distinguish a machine-generated utterance from a human-generated utterance, since the machine response lacks basic commonsense [236, 259], personality [156, 255] and emotion [258]. Even worse, generation models are prone to produce a safe answer, such as "OK," "I don't understand what you are talking about," which can fit in almost all conversational contexts but would only hurt the user experience [103, 158]. Ke et al. [89] propose to explicitly control the function of the generated sentence, for example, for the same user query, the system can answer with different tones: The interrogative tone can be used to acquire further information; the imperative tone is used to make requests, directions, instructions or invitations to elicit further interactions; and the declarative tone is commonly used to make statements or explanations. Another problem is how to evaluate the generated response, since there is no standard answer; we will further discuss this in Section 6.

Researchers borrow the ideas from dialog systems and apply the technologies in the user inferface of CRSs. For instance, Li et al. [107] generate responses by a decoder where a GRU model [32] decodes the context from the previous component (i.e., predicted sentiment towards items) to predict the next utterance step by step. Liu et al. [119] adopt the responding model in the work of Wu et al. [222] and propose both a retrieval-based model and a generation-based model to produce responses in their CRS.

However, a correct sentence does not mean it can fulfill the task of recommendation; at least the name of the recommended entity should be mentioned in generated sentences. Hence, Li et al. [107] use a switch to decide whether the next predicted word is a movie name or an ordinal word; Liu et al. [119] introduce an external memory module for storing all related knowledge, making the models select appropriate knowledge to enable proactive conversations. Besides, there are other efforts to guarantee the generated responses should not only be proper and accurate but also be meaningful and useful.

4.2.2. *Incorporating Recommendation-oriented Information*

There is a major limitation CRSs that use the end-to-end frameworks as the user interface: only items mentioned in the training corpus have a chance of being recommended since items that have never been mentioned are not modeled by the end-to-end model. Therefore, the performance of this method is greatly limited by the quality of human recommendations in the training data. To overcome this problem, Chen et al. [28] propose to incorporate domain knowledge to assist the recommendation engine. The incorporation of a knowledge graph mutually benefits the dialogue interface and the recommendation engine in the CRS. (1) the dialogue interface can help the recommender engine by linking related entities in the knowledge graph; the recommendation model is based on the R-GCN model [171] to extract information from the knowledge graph; (2) the recommender system can also help the dialogue interface: by mining words with high probability, the dialogue can connect movies with some biased vocabularies, thus it can produce consistent and interpretable responses.

Following this line, Zhou et al. [261] point out the remaining problems in the dialogue interface in CRSs. Although Chen et al. [28] have introduced an item-oriented knowledge graph to enable the system to understand the movie-related concepts, the system still cannot comprehend some words in the raw utterances. For example, "thriller", "scary", "good plot". In essence, the problem originates from the fact that the dialog component and the recommender component correspond to two different semantic spaces, namely word-level and entity-level semantic spaces. Therefore, Zhou et al. [261] incorporate and fuse two special knowledge graphs, i.e., a word-oriented graph (ConceptNet [181]), and an item-oriented graph (DBpedia [10]), to enhance understanding semantics in both the components. The representations of the same concepts on the two knowledge graphs are forced to be aligned with each other via the mutual information maximization (MIM) technique [197, 234]. Furthermore, a self-attention-based recommendation model is proposed to learn the user preference and adjust the representation of corresponding entities on the knowledge graph. Then, equipped with these representations containing both semantics and users' historical preferences, the authors use an encoder-decoder model to extract user intention from the raw utterances and directly generate the responses containing recommended items.

Besides, some researchers try to improve the diversity or explainability of generated responses in CRSs. For example, Liu et al. [119] propose the multi-topic learning that can handle diverse dialogue types in CRSs. To enhance the interpretability of CRSs, Chen et al. [30] design an incremental multi-task learning framework to integrate review comments as side information. Hence, the CRS can simultaneously produce a recommendation as well as a sentence as an explanation, e.g., "I recommend *Mission Impossible*, because it is by far the best of the action series." Moreover, Luo et al. [124] use a VAE-based architecture to learn a latent representation for generating recommendations and fitting user critiquing. Therefore, their model can better understand users' intentions from users' raw comments, and thus can generate more interpretable responses. Gao et al. [56] consider attributes and review information and rewrite a coherent and meaningful answer from a selected prototype answer, which can address the safe answer problem in the response [103, 158].

4.3. Section Summary

In Table 4, we classify CRSs into two classes in terms of the forms of input and output. Generally, interactive recommendations [270, 204, 240, 44], critiquing methods [26, 219, 124, 123], and CRSs focusing on the multi-turn conversation strategy [35, 34, 98, 100, 108] are prone to use the

Table 4
Mechanisms of language understanding and generation in CRSs.

Forms of Input & Output	Publications
Pre-annotated Input & Template-based Output	[253, 268, 120, 245, 187], [35, 34, 98, 100, 108, 51]
Raw Language Input & Natural Language Generation	[162, 107, 28], [261, 127, 119]

pre-annotated input and rule-based or template-based output; dialogue systems [236, 259, 56] and CRSs caring about the dialogue ability [107, 28, 261] are more likely to use raw natural language as input and automatically generate responses. In the future, user understanding and response generation in CRSs will remain a critical research field, as they serve as the interface of CRSs and directly impact the user experience.

5. Exploration-Exploitation Trade-offs

One challenge of CRSs is to handle the cold-start users that have few historical interactions. A natural way to tackle this is through the idea of the Exploration-Exploitation (E&E) trade-off. With exploitation, the system takes advantage of the best option that is known; with exploration, the system takes some risks to collect information about unknown options. In order to achieve long-term optimization, one might make a short-term sacrifice. In the early stages of E&E, an exploration trial could be a failure, but it warns the model to not take that action too often in the future. Although the E&E trade-off is mainly used for the cold-start scenario in CRSs, it can also be used for improving the recommendation performance for any users (including cold users and warm-up users) in recommendation systems.

MAB is a classic problem formulated to illustrate the E&E trade-off, and many algorithms have been proposed to solve the problem. In CRSs, the MAB-based algorithms are introduced to help the system improve its recommendation. Besides, there are also CRSs that use meta-learning to balance E&E. We first introduce MAB and common MAB-based algorithms in recommender systems, then we present examples how CRSs balance E&E in their models.

5.1. Multi-Armed Bandits In Recommendation

We first introduce the general MAB problem and the classic methods to solve it, then we introduce how recommender systems use MAB-based methods to achieve the E&E balance.

5.1.1. Introduction to Multi-Armed Bandits

MAB is a classic problem that well demonstrates the E&E dilemma [87, 4]. The name comes from the story where a gambler at a row of slot machines (each of which is known as a "one-arm bandit") wants to maximize his expected gain and has to decide which machines to play, how many times to play each machine, in which order to play them, and whether to continue with the current machine or try a different machine. The problem is difficult because all of the slot machines are black boxes, whose properties, i.e., the probability of winning, can only be estimated by the rewards observed in previous experiments.

Formally, the problem is to maximize the cumulative reward $\sum_{t=1}^{T} r_{a,t}$ after T rounds of arm selection. Here, $r_{a,t}$ is the reward with arm $0 \leq a \leq K$ selected at trial t, K is the total number of arms. Figure 6 illustrates an example in which a gambler decides which arm to choose now. For a certain arm, a reward distribution is estimated based on previous experiment results. The gambler can, naturally, select to exploit the second arm which has the maximal mean reward $\mu(a)$. Or, he can take some risks to explore the other arm, e.g., the third arm, which has a higher uncertainty $\Delta(a)$ and thus has the maximal upper confidence bound (UCB) of the reward $\mu(a) + \Delta(a)$. After each time he plays an arm, the new reward value is observed, and the estimated reward distribution of this arm can be updated accordingly. With exploration, the gambler hopes to find the potential arms that have higher rewards, though it can also end up in lower rewards. In any case the gambler has a better estimation of the rewards of those arms.

Equivalently, the problem can also be formulated as minimizing the regret function, which is the difference between the theoretically optimal expected cumulative reward and the estimated expected cumulative reward:

$$\mathbf{E}\left[\sum_{t=1}^{T} r_{t,a^*}\right] - \mathbf{E}\left[\sum_{t=1}^{T} r_{t,a}\right], \tag{5}$$

where a^* is the theoretically optimal arm with the maximum expected reward at all times.

The commonly used bandit strategies include the greedy strategy, i.e., the exploit-only strategy that always selects the arm with the current estimated highest reward; the random strategy, i.e., a trivial explore-only strategy; and ϵ-greedy, which mixes the greedy and random strategies via a trigger

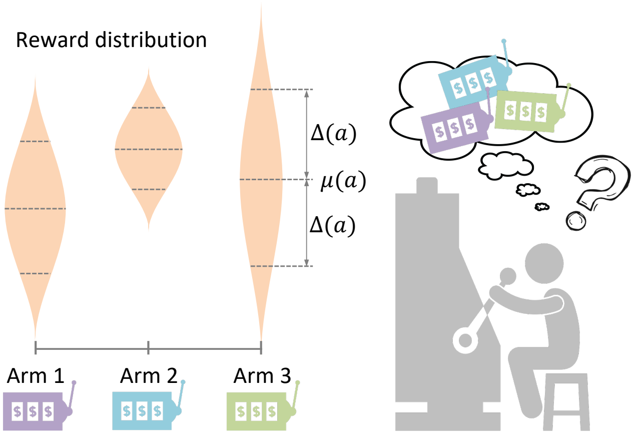


Figure 6: An illustration of the multi-armed bandit problem.

with probability ϵ. Other classic models include Upper Confidence Bound (UCB) [3, 4] and Thompson Sampling (TS) [21] which are introduced next.

5.1.2. Recommendation via MAB-based Methods

As the classic algorithm for E&E trade-offs, MAB-based models can be seamlessly plugged into the online recommendation setting [239, 254], interactive recommendation [253, 205], and CRSs [35, 243, 108]. In the online or interactive recommendation tasks, the system aims to recommend the optimal item(s) according to users' previous feedback. This process can be deemed as a MAB problem, where each arm corresponds to an item. Therefore, the classical MAB-based methods can be plugged in this situation.

However, traditional bandit methods only consider treating items as independent arms and ignore the item features [105]. Directly estimating each item's probability of being chosen based on the accumulated rewards is rather inefficient due to a large number of items. In recommendation, there is a rich set of features on users and items, and whether a user u_t would choose item a_t can be predicted by the features of both u_t and a_t. Motivated by this, Li et al. [105] propose a linear contextual bandit model called LinUCB, which is the first bandit model that considers the contextual information (i.e., user/item features) in recommendation systems.

For each trial t, they assume the expected reward r_t of a user u_t choosing an arm (item) a_t is linear in its d-dimensional feature vector $\mathbf{x}_{u_t,a_t}$ with the unknown coefficient vector θ_a^* (which is determined on this arm a_t rather than other arms); namely, for all trial t,

$$\mathbf{E}\left[r_{t,a} \mid \mathbf{x}_{u_t,a_t}\right] = \mathbf{x}_{u_t,a_t}^{\top}\boldsymbol{\theta}_a^*, \tag{6}$$

where the feature vector $\mathbf{x}_{u_t,a_t}$ summarizes information of both user u_t and arm (item) a_t, and is referred to as the context. The coefficients $\boldsymbol{\theta}_a^*$ can be learned from the historical interactions and feedback. Specifically, let $\mathbf{D}_a$ be a design matrix of dimension $m \times d$ at trial t, e.g., m contexts that are observed previously for arm a, and $\mathbf{c}_t \in \mathbb{R}^m$ be the corresponding reward vector, the coefficients $\boldsymbol{\theta}_a^*$ are estimated by applying ridge regression to the training data $\left(\mathbf{D}_a, \mathbf{c}_a\right)$ as:

$$\hat{\boldsymbol{\theta}}_a = \left(\mathbf{D}_a^{\top}\mathbf{D}_a + \mathbf{I}_d\right)^{-1}\mathbf{D}_a^{\top}\mathbf{c}_a,$$

where $\mathbf{I}_d$ is the $d \times d$ identity matrix. When components in $\mathbf{c}_a$ are independent conditioned on corresponding rows in $\mathbf{D}_a$, it can be shown that with probability at least $1-\delta$,

$$\left|\mathbf{x}_{u_t,a_t}^{\top}\hat{\boldsymbol{\theta}}_a - \mathbf{E}\left[r_{t,a} \mid \mathbf{x}_{u_t,a_t}\right]\right| \le \alpha\sqrt{\mathbf{x}_{u_t,a_t}^{\top}\left(\mathbf{D}_a^{\top}\mathbf{D}_a + \mathbf{I}_d\right)^{-1}\mathbf{x}_{u_t,a_t}},$$

for any $\delta > 0$ and $\mathbf{x}_{u_t,a_t} \in \mathbb{R}^d$, where $\alpha = 1 + \sqrt{\ln(2/\delta)/2}$ is a constant. Therefore, the inequality gives a reasonably tight UCB for the expected reward of arm a_t, from which the arm-selection (recommendation) strategy can be derived: at each trial t, choose

$$a_t \overset{\text{def}}{=} \arg\max_{a\in\mathcal{A}_t}\left(\mathbf{x}_{u_t,a_t}^{\top}\hat{\boldsymbol{\theta}}_a + \alpha\sqrt{\mathbf{x}_{u_t,a_t}^{\top}\left(\mathbf{D}_a^{\top}\mathbf{D}_a + \mathbf{I}_d\right)^{-1}\mathbf{x}_{u_t,a_t}}\right).$$

Actually, the contextual bandit model improves the recommendation by leveraging the user/item features through the idea of collaborative filtering [168, 169], i.e., those items are more likely to be recommended to a user who showed preference for items with similar features.

There are also studies pointing out that exploration in recommendations is important, i.e., the recommendations should be diverse instead of being limited by similar items [157, 118, 44]. For instance, Ding et al. [44] consider the fact that users may have different preference with regard to the diversity of items, e.g., a user with specific interest may prefer a relevant item set than a diverse item set, while another user without specific interest may prefer a diverse item set to explore his interests. Therefore, the authors propose a bandit learning framework to consider the user's preferences on both the item relevance features and the diversity features. It is a way to trade off the accuracy and diversity of recommendation results.

Besides, Yu et al. [238] use a cascading bandit in a visual dialog augmented interactive recommender system. In cascading bandits, the user examines the recommended list from the first item to the last and selects the first attractive one [94, 267]. This setting is practical to implement in online recommender systems or search engines. It has an excellent advantage as it can provide reliable negative samples, which are critical for recommendation, and the problem has drawn a lot of research attention [25, 43, 216, 110, 25]. Since the system can ensure that the items before the first selected one are not attractive, thus it can easily obtain reliable negative samples. Another contribution is the use of the item's visual appearance and user feedback to design more efficient exploration.

In addition, there are other efforts to enhance bandit methods in different recommendation scenarios. For instance, Chou et al. [33] indicate that a user would only choose one or a few arms in the candidates, leaving out the informative non-selected arms. They propose the concept of pseudo-rewards, which embeds estimates to the hidden rewards of non-selected actions under the bandit setting. Wang et al. [208] consider dependencies among items and explicitly formulate the item dependencies as clusters on arms, where arms within a single cluster share similar latent topics. They adopt a generative process based on a topic model to explicitly formulate the arm dependencies as the clusters on arms, where dependent arms are assumed to be generated from the same cluster. Yang et al. [231] consider the situations where there are exploration overheads, i.e., there are non-zero costs associated with executing a recommendation (arm) in the environment, and hence, the policy should be learned with a fixed exploration cost constraint. They propose a hierarchical learning structure to address the problem. Sakhi et al. [166] state that the online bandit signal is sparse and uneven, so they utilize the massive offline historical data. The difficulty is that most of offline data is irrelevant to the recommendation task, and the authors propose a probabilistic model to solve it.

The advantage of multi-armed bandit methods is their ability

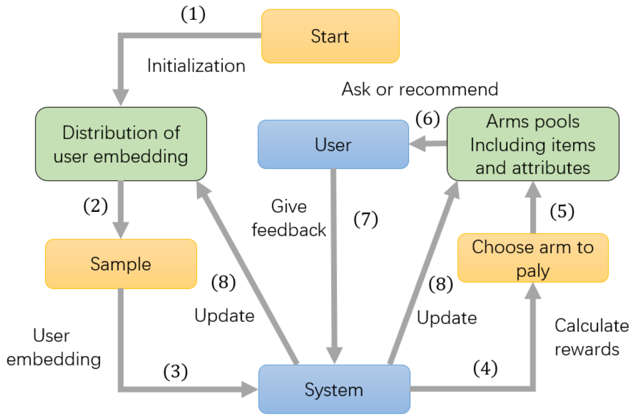


Figure 7: The flowchart of the ConTS algorithm. Credits: Li et al. [108].

to conduct online learning, enabling the model to learn the preferences of cold users and adjust the strategy quickly after several trials to pursue a global optimum.

5.2. Multi-Armed Bandits in CRSs

The ability to interact with users enables CRSs to directly use MAB-based methods to help the recommendation. Christakopoulou et al. [35] propose a classic CRS based on MAB, which uses several naive MAB-based methods to enhance the offline probabilistic matrix factorization (PMF) model [167]. They first initialize the model parameters using offline data, then leverage real-time user feedback to update parameters via several common multi-armed bandit models, including the aforementioned greedy strategy, random strategy, UCB [3, 4], and TS [21]. On the one hand, the performance improves on the initialized model due to the online updating; on the other hand, the offline initialization helps bandit methods reduce the computational complexity.

As mentioned above, the original MAB methods ignore item features, which could be very helpful in recommendation. Hence, Zhang et al. [243] propose a conversational upper confidence bound (ConUCB) algorithm to apply the LinUCB model [105] in the CRS context. Instead of asking items, ConUCB asks the user about one or more attributes (key-terms in their work). Specifically, they make the assumption that user preference on attributes can propagate to items, hence the system can analyze user feedback on queried attributes to quickly narrow down the item candidates. The strategies to select the attributes and arms depend on both the attribute-level and arm-level rewards, i.e., the feedback on attributes and items will be absorbed into the model parameters for future use. In addition, the authors employ a hand-crafted function to determine the timing to ask attributes or make recommendation, e.g., making k conversations in every m rounds.

However, hand-crafted strategies are fragile and inflexible, as the system should make recommendation only when the confidence is high. Therefore, Li et al. [108] propose a Conversational Thompson Sampling method (ConTS) to automatically alternate asking questions about attributes with recommending items. They achieve this goal by unifying all attributes and items in the same arm pool, thus an arm selected from the arm pool can be either a recommendation about an item or a question about an attribute. The flowchart of ConTS is illustrated in Figure 7. ConTS assumes each user's preference vector $\tilde{\mathbf{u}}$ is sampled from a prior Gaussian distribution as $\tilde{\mathbf{u}} \sim \mathcal{N}\left(\boldsymbol{\mu}_u, l^2\mathbf{B}_u^{-1}\right)$, where the $\boldsymbol{\mu}_u$, $\mathbf{B}$, and l are parameters.

For each new-coming user, the mean of prior Gaussian distribution, $\boldsymbol{\mu}_u$, is initialized by the average of existing users' preference vector $\mathcal{U}^{\text{old}}$ as:

$$\boldsymbol{\mu}_u = \frac{1}{|\mathcal{U}^{\text{old}}|}\sum_{i=1}^{|\mathcal{U}^{\text{old}}|}\mathbf{u}_i, \mathbf{u}_i \in \mathcal{U}^{\text{old}}. \tag{7}$$

The expected reward of arm a (which can either be an item or an attribute) for user u is also formulated as a Gaussian distribution since the Gaussian family is conjugate to itself. The expected reward is written as:

$$r\left(a, u, \mathcal{P}_u\right) \sim \mathcal{N}\left(\tilde{\mathbf{u}}^{\top}\mathbf{x}_a + \sum_{p_i \in \mathcal{P}_u}\mathbf{x}_a^T\mathbf{p}_i, l^2\right), \tag{8}$$

where $\mathcal{P}_u$ denotes the user's currently known preferred attributes obtained in historical conversations. And x_a represents the embedding vector of an arm. In the reward function, the term $\tilde{\mathbf{u}}^{\top}\mathbf{x}_a$ models the general preference of user u to arm a, and the term $\sum_{p_i \in \mathcal{P}_u}\mathbf{x}_a^T\mathbf{p}_i$ models the affinity between arm a and the user's preferred attributes $\mathcal{P}_u$. Then ConTS select an arm with the maximal reward as:

$$a(t) = \text{argmax}_{a \subset \mathcal{A}_u}\tilde{\mathbf{u}}^{\top}\mathbf{x}_a + \sum_{p_i \in \mathcal{P}_u}\mathbf{x}_a^T\mathbf{p}_i. \tag{9}$$

Note that if the $a(t)$ is an attribute, the system will query the user about the preference on this attribute; if it is an item, the system will make a recommendation using this item. After obtaining users' feedback, parameters such as $\boldsymbol{\mu}_u, \mathcal{P}_u, \mathbf{B}$ will be updated accordingly.

5.3. Meta Learning for CRSs

Beyond multi-armed bandits, there are work trying to balance between exploration and exploitation via meta learning. For instance, Zou et al. [271] formulate the interactive recommendation as a meta-learning problem, where the objective is to learn a learning algorithm that takes the user's historical interactions as the input and outputs a model (policy function) that can be applied to new users. The authors follow the idea of meta reinforcement learning [47] and use Q-Learning [139] to learn the recommendation policy. The exploration strategy is the aforementioned ϵ-greedy, where the model will select the items of maximum Q-value with probability $1-\epsilon$, and choose random items with probability ϵ.

In addition, Lee et al. [97] address the cold-start problem in recommendation via a model based on the Model-Agnostic Meta-Learning (MAML) algorithm [50]. The learned recommendation model can quickly adapt to the cold user

Table 5
E&E-based methods adopted by interactive recommender systems (IRSs) and CRSs.

	Mechanism	Publications
MAB in IRSs	Linear UCB considering item features	[105]
	Considering diversity of recommendation	[157, 118, 44]
	Cascading bandits providing reliable negative samples	[94, 267]
	Leveraging social information	[238]
	Combining offline data and online bandit signals	[166]
	Considering pseudo-rewards for arms without feedback	[33]
	Considering dependency among arms	[208]
	Considering exploration overheads	[231]
MAB in CRSs	Traditional bandit methods in CRSs	[35]
	Conversational upper confidence bound	[243]
	Conversational Thompson Sampling	[108]
	Cascading bandits augmented by visual dialogues	[238]
Meta learning for CRSs	Learning to learn the recommendation model	[97, 271, 217]

preference in the fine-tuning stage by asking the cold user a few questions about certain items (called the evidence candidates in the work). A drawback of this work is that the evidence candidates are only selected once, and the query process is conducted only at the beginning when cold users arrived. It could be better to extend this strategy to a CRS setting and develop a dynamic multi-round query strategy to further enhance the recommendation.

5.4. Section Summary

In this section, we introduce how a CRS can solve the cold-start problem and trade off the E&E balance via the interactive models such as MAB-based methods and meta learning methods. The solutions are summarized in Table 5. It still has a lot of room for CRSs to develop potential models to address the E&E problem, in order to improve the user experience.

6. Evaluation and User Simulation

In this section, we discuss how to evaluate CRSs, which is an underexplored problem. We group attempts to evaluate CRSs into two classes: (1) Turn-level evaluation, which evaluates a single turn of the system output, including the recommendation task and response generation task, which are both supervised prediction tasks. (2) Conversation-level evaluation, which evaluates the performance of the multi--turn conversation strategy which is a sequential decision making task. To achieve the goal, user simulation is important. We first introduce the commonly used datasets in CRSs, and then we introduce the metrics, methods, and problems in the turn-level and conversation-level evaluation of CRSs. Finally, we discuss the strategies of user simulation in CRSs.

6.1. Datasets and Tools

We list the statistics of the commonly used CRS datasets in Table 6. Some studies collect human-human and human-machine conversation data by asking true users to converse using natural language under certain rules. To guarantee the quality of the data, these users will be rewarded after providing qualified data. There are crowdsourcing sites, such as Amazon Mechanical Turk (AMT)[5], where the researchers can find participants to fulfill their data collection task [107, 141, 119, 64].

As mentioned earlier, a lot of studies of CRS focus on the interaction policy and the recommendation strategy instead of language understanding and generation. Thus, all these studies need is the labeled entities (including users, items, attributes, etc.) in the multi-turn conversation [245, 34, 98, 100, 108, 51]. These studies mainly simulate and construct the user interaction from the historical records in traditional recommendation datasets, e.g., MovieLens [9], LastFM [9], Yelp[6], and Amazon dataset [133].

Although it seems to be many datasets in CRSs, these datasets are not qualified to develop the CRSs that can work in industrial applications. The reason is twofold: first, the scale of these datasets is not enough to cover the real-world entities and concepts; second, the conversation is either constructed from the non-conversation data or generated under certain rigorous constraints, so it is hard to generalize to the complex and diverse real-world conversations. Therefore, more effort is needed to develop large-scale, generalizable, natural datasets for CRSs. Therefore, more effort is still needed to develop large-scale, generalizable, diverse, and natural datasets for CRSs.

There are many different settings in CRSs, making com-

[5] https://www.mturk.com/

[6] https://www.yelp.com/dataset

Table 6
Statistics of commonly used datasets of CRSs.

Dataset	#Dialogs	#Turns	Dialogue Type	Domains	Dialogue Resource	Related Publications
MovieLens [9]	Depended on the dialogue simulation process			Movie	From item ratings	[253, 120, 198, 271], [97, 77, 63]
LastFM [9]				Music	From item ratings	[98, 100, 262]
Yelp				Restaurant	From item ratings	[187, 98, 100]
Amazon [133]				E-commerce	From item ratings	[245, 52, 268, 152], [219, 124, 123, 51]
TG-ReDial [263]	10,000	129,392	Rec., chichat	Movie, multi topics	From item rating, and enhanced by multi topics	[263]
Facebook_Rec [45]	1M	6M	Rec.	Movie	From item ratings	[45]
COOKIE [52]	Not given	11,638,418	Rec.	E-commerce	From interactions and reviews on Amazon dataset [133]	[52]
HOOPS [51]	Not given	11,638,418	Rec.	E-commerce	From interactions and reviews on Amazon dataset [133]	[51]
DuRecDial [119]	10,190	155,477	Rec., QA, etc.	Movie, restaurant, etc.	Generated by workers	[119]
OpenDialKG [141]	15,673	91,209	Rec. chitchat	Movie, book, sport, etc.	Generated by workers	[141]
ReDial [107]	10,006	182,150	Rec., chitchat	Movie	Generated by workers	[107, 28, 261, 127]
MGConvRex [225]	7.6K+	73K	Rec.	Restaurant	Generated by workers	[225]
GoRecDial [86, 127]	9,125	170,904	Rec.	Movie	Generated by workers	[86]
INSPIRED [64]	1,001	35,811	Rec.	Movie	Generated by workers	[64]
ConveRSE [76]	Not given	9,276	Rec.	Movie, books, music	Generated by workers	[76, 77]

parison between different models difficult. Recently, Zhou et al. [260] have implemented an open-source toolkit, called CRSLab[7], for building and evaluating CRSs. They unify the tasks in existing CRSs into three sub-tasks: namely recommendation, conversation and policy, which correspond to our three components in Figure 3: recommendation engine, user interface, and conversation strategy module, respectively. Some models and metrics are implemented under the three tasks, and the toolkit contains an evaluation module that can not only conduct the automatic evaluation but also the human evaluation through an interaction interface, which makes the evaluation of CRSs more intuitive. However, up to now, the majority of implemented methods are based on end-to-end dialogue systems [107, 28, 261] or deep language models [263]; the CRSs that focus on the interaction policy and the multi-turn conversation strategies ([100, 98]) are absent.

6.2. Turn-level Evaluation

The fine-grained evaluation of CRSs is conducted on the output of each single turn, which contains two tasks: language generation and recommendation.

6.2.1. Evaluation of Language Generation

For CRS models that generate natural language-based responses to interact with users, the quality of the generated responses is critical. Thus we can adopt the metrics used in dialogue response generation to evaluate the output of CRS. Two example metrics are BLEU [147] and Rouge [113]. BLEU measures the precision of generated words or n-grams compared to the ground-truth words, representing how much the words in the machine-generated utterance appeared in the ground-truth reference utterance. Rouge measures the recall of it, i.e., how many of the words or n-grams in the ground-truth reference utterance appear in the machine-generated utterance.

However, it is widely debated whether these metrics are suitable for evaluating language generation [114, 145]. Because those metrics are only sensitive to lexical variation, they cannot appropriately assess semantic or syntactic variations of a given reference. Meanwhile, the goal of the proposed system is not to predict the highest probability response, but rather the long-term success of the dialogue. Thus, other metrics reflecting user satisfaction are more suitable in evaluation, such as measuring fluency [17, 142, 46], consistency [53, 96], readability [95], informativeness [74], diversity [104, 78, 57], and empathy [58, 175]. For more metrics and evaluation methods on text generation, we refer the readers to the overviews [18, 41].

However, the CRSs based on end-to-end dialogue frameworks or deep language models may have limitations regarding the usability in practice. Recently, Jannach and Manzoor [81] conducted an evaluation on the two state-of-the-art end-to-end frameworks [107, 28], and showed that both models face three critical issues: (1) For each system, about one-third of the system utterances are not meaningful in the given context and would probably lead to a breakdown of the

[7] https://github.com/RUCAIBox/CRSLab

conversation in a human evaluation. (2) Less than two-thirds of the recommendations were considered to be meaningful in a human evaluation. (3) Neither of the two systems "generated" utterances, as almost all system responses were already present in the training data. Jannach and Manzoor [81]'s analysis shows that human assessment and expert analysis are necessary for evaluating CRS models as there is no perfect metric to evaluate all aspects of a CRS. The CRS models and their evaluation still have a long way to go.

6.2.2. Evaluation of Recommendation

The performance of recommendation models is evaluated by comparing the predicted results with the records in the test set. There are two kinds of metrics in measuring the performance of recommender systems:

- **Rating-based Metrics.** These metrics assume the user feedback is an explicit rating score, e.g., an integer in the range of one to five. Therefore, we can measure the divergence between the predicted scores of models and the ground-truth scores given by users in the test set. Conventional rating-based metrics include Mean Squared Error (MSE) and Root Mean Squared Error (RMSE), where RMSE is the square root of the MSE.
- **Ranking-based Metrics.** These metrics are more frequently used than rating-based metrics. Ranking-based metrics require that the relative order of predicted items should be consistent with the order of items in the test set. Thereby, there is no need for explicit rating scores from users, and the implicit interactions (e.g., clicks, plays) can be used to evaluate models. For example, a good evaluation result means that the model should only recommend the items in the test set, or it means that the items with higher scores in the test set should be recommended at higher ranks than the items with lower scores. Frequently used ranking-based metrics include hits, precision, recall, F1-score, Mean Reciprocal Rank (MRR), Mean Average Precision (MAP), and Normalized Discounted Cumulative Gain (NDCG) [83].

  Recently, it has become common for researchers to speed up evaluation by sampling a small set of of irrelevant items and calculate the ranking-based metrics only on the small set [67, 48, 71, 230]. However, Krichene and Rendle [92] point out and prove that some metrics, such as average precision, recall, and NDCG, are inconsistent with the exact metrics when they are calculated on the sampled set. This means that if a recommender A outperforms a recommender B on the sampled metric, it does not imply that A has a better metric than B when the metric is computed exactly. Therefore, the authors suggest that sampling during evaluation should be avoided; if it is necessary to sample, using the corrected metrics proposed by the authors is a better choice.

The biggest problem in these evaluation methods is that real-world user interactions are very sparse, and a large fraction of items never have a chance of being consumed by a user. However, this does not mean that the user does not like any of them. Perhaps the user has never seen them, or the user just does not have resources to consume them [115, 24]. Hence, taking the consumed items in the test set as the users' ground-truth preferences can introduce evaluation biases [230, 24]. Unlike static recommender systems, CRSs have the ability to ask real-time questions, so the system can make sure whether a user is satisfied with an item by collecting users' online feedback. This online user test can avoid biases and provide conversation-level assessments for the CRS model.

6.3. Conversation-level Evaluation

Different from the turn-level evaluation which compares the prediction results with the ground-truth labels in a supervised way, the conversation-level evaluation is not a supervised prediction task. The interaction process is not i.i.d. (independent and identically distributed) since each observation is part of a sequential process and each action the system makes can influence future observations. Plus, the conversation heavily relies on the user feedback. Therefore, the evaluation of the conversation requires either an online user test or leveraging historical interaction data which can be conducted by the off-policy evaluation or using user simulation.

6.3.1. Online User Test

The online user test, or A/B test, can directly evaluate the conversation policy by leveraging true user feedback. To conduct the assessment, the appropriate metrics should be designed. For example, the average turn (AT) is a global metric to optimize in a CRS, as the model should capture user intention and make successful recommendations thus finish the conversation with as few turns as possible [98, 100, 108]. A similar metric is the recommendation success rate (SR@t), which measures how many conversations have ended with the successful recommendation by the t-th turn. Besides, the ratio of failed attempts, e.g., how many of the questions asked by the system are rejected or ignored by users, can be a feasible way to measure whether a system makes decisions to the users' satisfaction.

Besides these global statistics, the cumulative performance of each turn of the conversation can also reflect the overall quality of the conversation. The expectation of the cumulative reward of a conversation policy can be written as:

$$J(\pi) = \mathbb{E}_{\tau \sim p_\pi(\tau)} \left[\sum_{t=0}^{T} \gamma^t r\left(\mathbf{s}_t, \mathbf{a}_t\right) \right], \tag{10}$$

where the conversation trajectory τ is a sequence of states and actions of length T, $p_\pi(\tau)$ is the trajectory distribution under policy π. $\gamma \in (0, 1]$ is a scalar discount factor. $r\left(\mathbf{s}_t, \mathbf{a}_t\right)$ is the immediate reward obtained by performing action $\mathbf{a}_t$ at state $\mathbf{s}_t$, e.g., it can be a feedback signal that reflects user satisfaction such as user clicks or dwell time [27, 75].

Though effective, the online user evaluation has critical problems: (1) The online interaction between humans and CRSs is slow and usually takes weeks to collect sufficient data to make the assessment statistically significant [106, 59, 251]. (2) Collecting users' feedback is expensive in terms of engineering and logistic overhead [80, 79, 227] and may hurt user experience as the recommendation may not satisfy them [172, 106, 59, 29]. Therefore, a natural solution is to leverage the historical interaction, where the off-policy evaluation and user simulation techniques can be used.

6.3.2. Off-Policy Evaluation

Off-policy evaluation, also called counterfactual reasoning or counterfactual evaluation, is designed to answer a counterfactual question: what would have happened if instead of π_β we would have used π_θ? Specifically, when we want to evaluate the current target policy π_θ but we only have data under a behavior policy (or logging policy) π_β, we can still evaluate the target policy π_θ by introducing the importance sampling or inverse propensity score [59, 80, 27, 135, 101] as:

$$J\left(\pi_\theta\right) = \mathbb{E}_{\tau\sim\pi_\beta(\tau)}\left[\frac{\pi_\theta(\tau)}{\pi_\beta(\tau)}\sum_{t=0}^{T}\gamma^t r(\mathbf{s},\mathbf{a})\right]. \qquad (11)$$

It is similar to Equation 10 except we use data logged under another policy to evaluate the target policy. Where a weight $w(\tau) = \frac{\pi_\theta(\tau)}{\pi_\beta(\tau)}$ is used to address the distribution mismatch between the two policy π_β and π_θ. Unfortunately, such an estimator suffers from high variance when π_θ deviates from π_β a lot. The variance reduction techniques are introduced as the remedy. The common techniques include weight clipping [27, 270] which limits $w(\tau)$ by an upper bound, and trusted region policy optimization (TRPO) [173, 27].

Another intuitive method is to directly simulate user behaviors just like the online user test, where user feedback is provided by the user simulators instead of true users. It is efficient and can avoid the high variance problem. However, the challenge is that the preference of simulated users may deviate from the true users, i.e., the user simulation can avoid high variance, but it introduces bias. Therefore, creating reliable user simulators is a crucial challenge.

6.3.3. User Simulation

There are generally four types of strategies in simulating users: (1) using the direct interaction history of users, (2) estimating user preferences on all items, (3) extracting from user reviews, and (4) imitating human conversational corpora.

- *Using Direct Interaction History of Users.* The basic idea is similar to the evaluation of traditional recommender systems, where a subset of human interaction data is set aside as the test set. If the items recommended by a CRS are in the users' test set, then this recommendation is deemed to be a successful one. As user-machine interactions are relatively rare, there is a need to generate/simulate interaction data for training and evaluation. Sun and Zhang [187] make a strong assumption that users visit restaurants after chatting with a virtual agent. Based on this assumption, they create a crowdsourcing task to use a schema-based method to collect dialogue utterances from the Yelp dataset. In total, they collect 385 dialogues, and simulate 875, 721 dialogues based on the collected dialogues by a process called delexicalization. For instance, "I'm looking for Mexican food in Glendale" is converted to the template: "I'm looking for <Category> in <City>", then they use these templates to generate dialogues by using the rating data and the rich information on the Yelp dataset. Lei et al. [98, 100] use click data in the LastFM and Yelp datasets to simulate conversational user interactions. Given an observed user-item interaction, they treat the item as the ground truth item to seek for and its attributes as the oracle set of attributes preferred by the user in this session. First, the authors randomly choose an attribute from the oracle set as the user's initialization to the session. The session goes into a loop of a "model acts – simulator responses" process, in which the simulated user will respond with "Yes" if the query entity is contained in the oracle set and "No" otherwise. Most CRS studies adopt this simulation method because of its simplicity [268, 34, 22]. However, the sparsity problem in recommender systems still remains: only a few values in the user-item matrix are known, while most elements are missing, which forbids the simulation on these items.
- *Estimating User Preferences on All Items.* Using direct user interactions to simulate conversations has the same drawbacks as we mentioned above, i.e., a large number of items that have not been seen by a user are treated as disliked items. To overcome this bias in the evaluation process, some research proposes to obtain the user preferences on all items in advance. Given an item and its auxiliary information, the key to simulating user interaction is to estimate or synthesize preferences for this item. For example, Christakopoulou et al. [35] ask 28 participants to rate 10 selected items, and then they can estimate the latent vectors of the 10 users' preferences based on their matrix factorization model. By adding noise to the latent vector, they simulate 50 new user profiles and calculate these new users' preferences on any items based on the same matrix factorization model. Zhang et al. [243] propose to use ridge regression to compute user preferences based on these known rewards on historical interaction and users' features; they synthesize the user's reaction (rewards) on each item according to the computed preferences. This kind of method can theoretically simulate a complete user preference without the exposure bias. However, because the user preferences are computed or synthesized, it could deviate from real user preferences. Huang et al. [73] analyze the phenomenon of popularity bias [182, 154] and selection bias [131, 68, 183] in simulators built on logged interaction data and try to alleviate model performance degradation due to these biases; it remains to be seen to which degree generated interactions of the type described above are subject to similar bias phenomena.
- *Extracting Information from User Reviews.* Besides user behavior history, many e-commerce platforms have textual review data. Unlike the consumption history, an item's review data usually explicitly mentions attributes, which can reflect the users' personalized opinions on this item.

Zhang et al. [245] transform each textual review of part of the Amazon dataset into a question-answer sequence to simulate the conversation. For example, when a user mentioned that a blue Huawei phone with the Android system in a review of a mobile phone X, then the conversation sequence constructed from this review is (Category: mobile phone → System: Android → Color: blue → Recommendation: X). Zhou et al. [263] also construct simulated interactions by leveraging user reviews. Based on a given user profile and its historical watching records, the authors construct a topic thread that consists of topics (e.g., "family" or "job seeking") extracted from reviews of these watched movies. The topic thread is organized by a rule and eventually leads to the target movie. And the synthetic conversation is fleshed out by retrieving the most related reviews under corresponding topics.

A noteworthy problem is that the aspects mentioned in reviews may contain some drawbacks of the products, which does not aid understanding why a user has chosen a product. For example, when a user complains about the capacity of a phone of 64 Gigabytes is not enough, and it should not be simply convert to (Storage capacity: 64 Gigabytes) for the CRS to learn. Thus, employing sentiment analysis on the review data is necessary, and only the attribute with positive sentiment should be considered as the reason in choosing the item [246, 248].

- *Imitating Humans' Conversational Corpora.* In order to generate conversational data without biases, a feasible solution is to use real-world two-party human conversations as the training data [196]. By using this type of data, a CRS system can directly mimic human behavior by learning from a large number of real human-human conversations. For example, Li et al. [107] ask workers from AMT to converse in terms of the topics on the movie recommendation. Using these conversational corpora as training data, the model can learn how to respond properly based on the sentiment analysis result. Liu et al. [119] conduct a similar data collection process. Except for collecting the dialogues about the recommendation, they also collect and construct a knowledge graph and define an explicit profile for each worker who seeks recommendations. Therefore, the conversational topics can contain many non-recommendation scenarios, e.g., question answering or social chitchat, which are more common in real life. To evaluate this kind of model, besides considering whether the user likes the recommended item, we have to consider if the system responds properly and fluently. The BLEU score [148] is used to measure the fluency of these models mimicking human conversations [12, 247].

  There are also drawbacks for this kind of method. First, when collecting the human conversational corpus, two workers need to enter the task at the same time, which is a rigorous setting and thus limits the scale of the dataset. Second, designers usually have many requirements that restrict the direction of the conversation. Therefore, the generated conversation is constrained and cannot fully cover the real-world scenarios. By imitating a collected corpus, learning a conversation strategy is very sensitive to the quality of the collected data. Vakulenko et al. [196] analyze the characteristics of different human-human corpora, e.g., in terms of initative taking, and show that there are important differences between human-human and human-machine conversations.

Recently, Zhang and Balog [241] have investigated using user simulations in evaluating CRSs. They organize the action sequence of the simulated user as a stack-like structure, called the user agenda. A dynamic update of the agenda is regarded as a sequence of pull or push operations, where dialogue actions are removed from or added to the top. Figure 8 shows an example of a dialogue between the simulated user and a CRS. At each turn, the simulated user updates its agenda by either a push or a pull operation based on the dialogue state and the CRS's action. The authors define a set of actions and the transition rule on these actions to let the simulated user imitate real users' intentions. For example, the *Disclose* action indicates that the user expresses its need either actively, or in response to the agent's question, e.g., "I would like to arrange a holiday in Italy". And after this action, the simulator can either transit to the *Inquire* action or the *Reveal* section based on how the CRS model acts.

Besides modeling the user preference in simulation, another branch of studies considers modeling user behaviors in the slate, top-K, or list-wise recommendation. A natural solution is to consider the combinatorial action which contains a list of items instead of a single item [188]. However, this method is unable to scale to problems of the size encountered in large, real-world recommender systems. The feasible way is to assume a user only consumes a single item from each slate and the obtained reward only depends on the item [75]. Under the assumption, user choice behavior can be modeled as the multinomial logit model [121] or the cascade model [75, 238, 94, 267].

Despite the recent interest in developing reliable user simulators, we believe that the research in this field is in its infancy and needs a lot of advancements.

6.4. Section Summary

In this section, we review the metrics, methods, and challenges in the turn-level evaluation and conversation-level evaluation of CRSs. The turn-level evaluation measures the performance of the supervised prediction tasks, i.e., recommendation and language generation of the CRS in a single round; the conversation-level evaluation measure how the conversation strategy performs in the multi-turn conversation. Since an online user test is expensive to conduct, researchers either leverage the off-policy evaluation which assesses the target policy using the logged data under the behavior policy, or directly introduce user simulators to replace the true users in evaluation.

The evaluation of CRSs still needs a lot of effort. It ranges from constructing large-scale dense conversational recommendation data, to proposing uniform evaluation methods to compare different CRS methods that integrate both recommendation and conversation aspects.

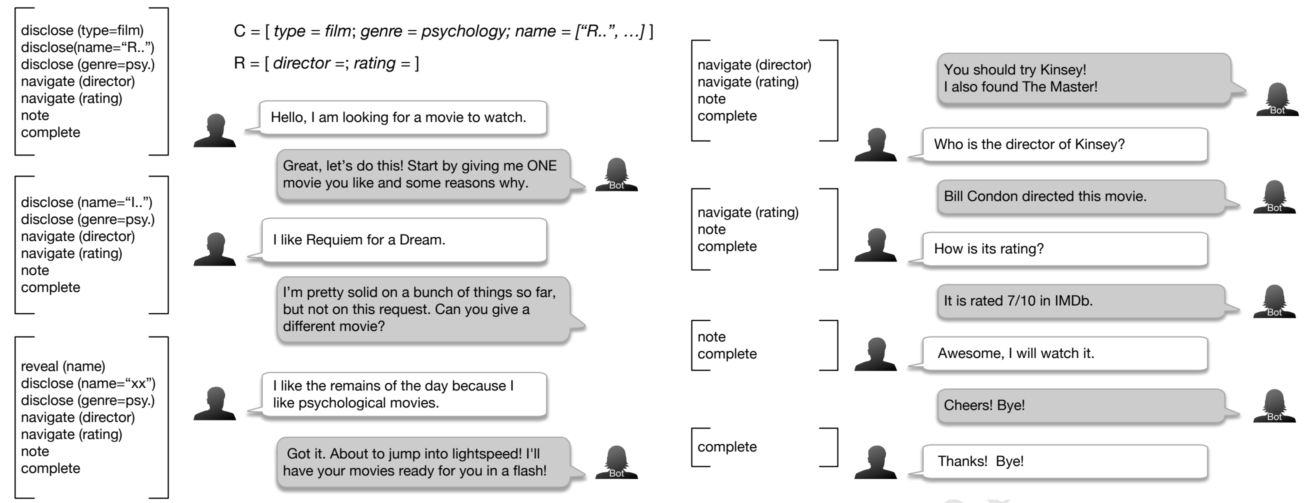

disclose (type=film)
disclose(name="R..")
disclose (genre=psy.)
navigate (director)
navigate (rating)
note
complete
C = [type = film; genre = psychology; name = ["R..", ...]]
R = [director =; rating =]
Hello, I am looking for a movie to watch.
Great, let's do this! Start by giving me ONE movie you like and some reasons why.
Bot
disclose (name="I..")
disclose (genre=psy.)
navigate (director)
navigate (rating)
note
complete
I like Requiem for a Dream.
I'm pretty solid on a bunch of things so far, but not on this request. Can you give a different movie?
reveal (name)
disclose (name="xx")
disclose (genre=psy.)
navigate (director)
navigate (rating)
note
complete
I like the remains of the day because I like psychological movies.
Got it. About to jump into lightspeed! I'll have your movies ready for you in a flash!
Bot
navigate (director)
navigate (rating)
note
complete
You should try Kinsey! I also found The Master!
Bot
Who is the director of Kinsey?
navigate (rating)
note
complete
Bill Condon directed this movie.
Bot
How is its rating?
It is rated 7/10 in IMDb.
Bot
note
complete
Awesome, I will watch it.
Cheers! Bye!
Bot
complete
Thanks! Bye!

Figure 8: Example dialogue with agenda sequence and state transition. The agenda is shown in square brackets. The third agenda is a result of a push operation, all other agendas updates are pull operations. Credits: Zhang and Balog [241].

7. Future Directions and Opportunities

Having described key advances and challenges in the area CRSs, we now envision some promising future directions.

7.1. Jointly Optimizing Three Tasks

The recommendation task, language understanding and generation task, and conversation strategies in CRSs are usually studied separately in the three components in Figure 3, respectively. The three components share certain objectives and data with each other [28, 127, 98, 261]. For example, the user interface feeds extracted aspect-value pairs to the recommendation engine, and then integrates the entities produced by the recommendation engine into the generated response. However, they have the exclusive data that does not benefit each other. For instance, the user interface may use the rich semantic information in reviews but not shares with a recommendation engine [107]. Besides, the two components may work in the end-to-end framework that lacks an explicit conversation strategy to coordinate them in the multi-turn conversation [107, 28], thus the performance is not satisfied in human evaluation [81].

Thereby, the three tasks should be jointly learned and guided by an explicit conversation strategy for their mutual benefit, for instance, what if the conversation strategy module were able to plan future dialogue acts based on item-item relationships (such as complementarity and substitutability [132, 201, 116])?

7.2. Bias and Debiasing

It is inevitable that a recommender system could encounter various types of bias [24]. Some types of biases, e.g., popularity bias [1, 182] and conformity bias [248, 117], can be removed with introducing the interaction between the user and system. For example, a static recommender may not be sure whether a user will follow the crowd and like popular items. Therefore, the popularity bias is introduced in the recommender system since popular items can have higher probability of being recommended. This, however, could be avoided in CRSs because a CRS can query about the user's attitude towards popular items in real time and avoid recommending them if the user gives negative feedback.

Nevertheless, some types of bias persist. For example, even though a recommender system may provide access to a large number of items, a user can only interact with a small set of them. If these items are chosen by a model or a certain exposure mechanism, users have no choices but to keep consuming these items. That is the exposure bias [115]. Moreover, users often select or consume their liked items and ignore these disliked ones even these items have been exposed to users, which introduces the selection bias [131, 68, 183], also known as the positivity bias [73, 154], i.e., rating data is often missing not at random and the missing ones are more likely to be disliked by the user [68]. These types of bias can be amplified in the feedback loop and may hurt the recommendation model [178, 185]. For instance, a CRS model polluted by biased data might repeatedly generate the same items even through users suggested they would like other ones.

There are relatively few efforts to study the bias problem in CRSs. The exploration-exploitation methods introduced in Section 5 can alleviate some types of bias in CRSs. And Huang et al. [73] make an attempt to remove the positivity bias in the user simulation stage for the interactive recommendation. Moreover, Chen et al. [24] present a comprehensive survey of different types of bias and describe a number of debiasing methods for recommender systems (RSs); it provides some perspectives for debiasing CRSs.

7.3. Sophisticated Multi-turn Conversation Strategies

The multi-turn strategy considered in current studies of CRSs are relatively naive. For example, there is work using a hand-crafted function to determine the timing to ask attributes or make recommendation, e.g., making *k* conversations in every *m* rounds [243]. These studies based on end-to-end dialogue systems or deep neural language models are worse: they do not even have an explicit strategy to control the multi-turn conversation [107, 28]. Besides, some strategies can be problematic in regard to handling users' negative feedback. For instance, Lei et al. [98] consider updating the model parameters when the user dislikes a recommended item. However, simply taking rejected items as negative samples would influence the model's judgement on the queried attributes. For example, a user's rejection of a recreation video might be due to the fact that they watched it before, and it does not mean that they dislike recreation videos. To overcome this problem, the model should consider more sophisticated strategies such as recognizing reliable negative samples [25, 43, 216, 110, 25] as well as disentangling user preferences on items and attributes [126, 214].

We have witnessed some studies using RL as the multi-turn conversation strategy by determining model actions such as whether to ask or recommend [187, 98, 100]. However, there is a lot of room for improvement in designing the state, action, and reward in RL. For instance, more sophisticated actions can be taken into consideration such as answering open-domain questions raised by users [265] or chatting non-task-oriented topics for entertainment purposes [223, 119]. Besides, more advanced and intuitive RL technologies can be considered to avoid the difficulties, e.g., hard to train and converge, in basic RL models [206]. For example, Inverse RL (IRL) [144] can be considered to learn a proper reward function from the observed examples in certain CRS scenarios, where there are too many user behavior patterns so the reward is hard to define. Meta-RL [47, 207] can be adopted in CRSs, where the interaction is sparse and various, to speed up the training process and to improve the learning efficiency for novel subsequent tasks.

7.4. Knowledge Enrichment

A natural idea to improve CRSs is to introduce additional knowledge. In early stages of the development of CRSs, only the recommended items themselves were considered [35]. Later, the attribute information of items was introduced to assist in modeling user preferences [34]. Even more recent studies consider the rich semantic information in knowledge graphs [261, 100, 225, 141]. For example, to better understand concepts in a sentence such as "I am looking for scary movies similar to *Paranormal Activity (2007)*", Zhou et al. [261] propose to incorporate two external knowledge graphs (KGs): one word-oriented KG providing relations (e.g., synonyms, antonyms, or co-occurrence) between words so as to comprehend the concept "scary" in the sentence; one item-oriented KG carrying structured facts regarding the attributes of items.

Besides knowledge graphs, multimodal data can also be integrated into the original text-based CRSs since it can enrich the interaction from new dimensions. There are some studies that exploit the visual modality, i.e., images, in dialogue systems [238, 111, 37, 249]. For example, Yu et al. [238] propose a visual dialog augmented CRS model. The model will recommend a list of items in photos, and the user will give text-based comments as feedback. The image not only helps the model learn a more informative representation of entities, but also enable the system to better convey information to the user. Except for the visual modality, other modalities can benefit CRSs and could be integrated. For example, spoken natural language can convey users' emotions as well as sentiments towards certain entities [153].

7.5. Better Evaluation and User Simulation

The evaluation of CRSs still has a long way to go. As we introduced in Section 6.3, evaluating the CRS requires real-time feedback, which is expensive in real-world situations [80]. Thus, most CRSs adopt user simulation techniques to create an environment [241]. However, simulated users cannot fully replace human beings. How to simulate users with maximum fidelity still needs further research. Feasible directions include designing systematic simulation agenda [241, 170], building dense user interactions for reliable simulation [270, 29, 5], and modeling user choice behaviors over the slate recommendation [75, 135, 2].

In addition, CRSs work on different datasets and they have various assumptions and settings. Therefore, developing comprehensive evaluation metrics and procedures to assess the performance of CRSs remains an open problem. Recently, Zhou et al. [260] have implemented an open-source CRS toolkit, enabling evaluation between different CRS models. However, their implemented models are mainly based on end-to-end dialogue systems [107, 28, 261] or deep language models [263], the models focusing on the explicit conversation strategy [100, 98] are absent.

8. Conclusion

Recommender systems are playing increasingly important role in information seeking and retrieval. Despite having been studied for decades, traditional recommender systems estimate user preferences only in a static manner like through historical user behaviours and profiles. It offers no opportunities to communicate with users about their preferences. This inevitably suffers from a fundamental information asymmetry problem: a system will never know precisely what a user likes (especially when his/her preference drifts frequently) and why the user likes an item. The envision of Conversational recommender systems (CRSs) brings a promising solution to such problems. With the interactive ability as well as the natural language-based user interface, CRSs can dynamically get explicit user feedback using natural languages, while increasing user engagement and improving user experience. This bold vision provides great potential for the future of recommender system, hence actively contributes to the development of the next generation of information seeking techniques.

Although the build of CRS is an emerging field, we have spotted great efforts from different perspectives. In this survey, we acknowledge those efforts, with the aim to summarize the existing studies and to provide insightful discussions. We tentatively gave a definition of the CRS and introduced a general framework of CRSs that consists of three components: a user interface, a conversation strategy module and a recommender engine. Based on this decomposition, we distilled five existing research directions, namely: (1) question-based user preference elicitation; (2) multi-turn conversational recommendation strategies; (3) dialogue understanding and generation; (4) exploitation-exploration trade-offs for cold users; (5) evaluation and user simulation. For each direction, we reviewed the existing efforts and their limitation in one section, leading to the primary structure of this survey. Despite the progresses on the above five directions, more interesting problems remain to be explored in the field of CRSs, such as, (1) joint optimization of three components; (2) bias and debiasing methods in CRSs; (3) multi--turn conversational recommendation strategies; (4) multi–modal knowledge enrichment; (5) evaluation and user simulation.

Our discussions above provide a comprehensive retrospect of current progress of CRSs which can serve as the basis for the further development of this field. By providing this survey, we call arm to this emerging and interesting field. We hope this survey can inspire the researchers and practitioners from both industry and academia to push the frontiers of CRSs, making the thoughts and techniques of CRSs more prevalent for the next generation of information seeking techniques.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (U19A2079, 61972372) and the National Key Research and Development Program of China (2020AAA0106000).

All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

References

[1] Himan Abdollahpouri and Masoud Mansoury. 2020. Multi-sided exposure bias in recommendation. *International Workshop on Industrial Recommendation Systems (IRS2020) in Conjunction with ACM KDD '2020* (2020).

[2] M Mehdi Afsar, Trafford Crump, and Behrouz Far. 2021. Reinforcement Learning Based Recommender Systems: A Survey. *arXiv preprint arXiv:2101.06286* (2021).

[3] Peter Auer. 2002. Using Confidence Bounds for Exploitation-Exploration Trade-offs. *Journal of Machine Learning Research* 3, Nov (2002), 397–422.

[4] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. 2002. Finite-Time Analysis of the Multiarmed Bandit Problem. *Machine Learning* 47, 2-3 (2002), 235–256.

[5] Xueying Bai, Jian Guan, and Hongning Wang. 2019. Model-Based Reinforcement Learning with Adversarial Training for Online Recommendation. In *Advances in Neural Information Processing Systems (NeurIPS '19, Vol. 32)*.

[6] Vevake Balaraman and Bernardo Magnini. 2020. Proactive Systems and Influenceable Users: Simulating Proactivity in Task-oriented Dialogues. In *The 24th Workshop on the Semantics and Pragmatics of Dialogue (WatchDial '20)*.

[7] Ziv Bar-Yossef and Naama Kraus. 2011. Context-Sensitive Query Auto-Completion. In *Proceedings of the 20th International Conference on World Wide Web (WWW '11)*. 107–116.

[8] Robert Bell, Yehuda Koren, and Chris Volinsky. 2007. Modeling Relationships at Multiple Scales to Improve Accuracy of Large Recommender Systems. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '07)*. 95–104.

[9] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. 2011. The Million Song Dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR '2011)*.

[10] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. 2009. DBpedia - A Crystallization Point for the Web of Data. *Web Semantics: Science, Services and Agents on the World Wide Web* 7, 3 (Sept. 2009), 154–165.

[11] Craig Boutilier. 2002. A POMDP Formulation of Preference Elicitation Problems. In *Eighteenth National Conference on Artificial Intelligence (AAAI '02)*. 239–246.

[12] Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP '18)*. 5016–5026.

[13] Vanessa Buhrmester, David Münch, and Michael Arens. 2019. Analysis of Explainers of Black Box Deep Neural Networks for Computer Vision: A Survey. *arXiv preprint arXiv:1911.12116* (2019).

[14] Robin D. Burke, Kristian J. Hammond, and Benjamin C. Young. 1997. The FindMe Approach to Assisted Browsing. *IEEE Expert: Intelligent Systems and Their Applications* 12, 4 (July 1997), 32–40.

[15] Fei Cai and Maarten de Rijke. 2016. A Survey of Query Auto Completion in Information Retrieval. *Foundations and Trends in Information Retrieval* 10, 4 (September 2016), 273–363.

[16] Giuseppe Carenini, Jocelyin Smith, and David Poole. 2003. Towards More Conversational and Collaborative Recommender Systems. In *Proceedings of the 8th International Conference on Intelligent User Interfaces (IUI '03)*. 12–18.

[17] Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep Communicating Agents for Abstractive Summarization. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers) (NAACL '18)*. 1662–1675.

[18] Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2020. Evaluation of Text Generation: A Survey. *arXiv preprint arXiv:2006.14799* (2020).

[19] Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. 2020. Controllable Multi-Interest Framework for Recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '20)*. 2942–2951.

[20] Urszula Chajewska, Lise Getoor, Joseph Norman, and Yuval Shahar. 1998. Utility Elicitation as a Classification Problem. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI '98)*. 79–88.

[21] Olivier Chapelle and Lihong Li. 2011. An Empirical Evaluation of Thompson Sampling. In *Proceedings of the 24th International Conference on Neural Information Processing Systems (NIPS '11)*. 2249–2257.

[22] Haokun Chen, Xinyi Dai, Han Cai, Weinan Zhang, Xuejian Wang, Ruiming Tang, Yuzhou Zhang, and Yong Yu. 2019. Large-Scale Interactive Recommendation with Tree-Structured Policy Gradient. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI '19, Vol. 33)*. 3312–3320.

[23] 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.

[24] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2020. Bias and Debias in Recommender System: A Survey and Future Directions. *arXiv preprint arXiv:2010.03240* (2020).

[25] Jiawei Chen, Can Wang, Sheng Zhou, Qihao Shi, Yan Feng, and Chun Chen. 2019. SamWalker: Social Recommendation with Informative Sampling Strategy. In *The World Wide Web Conference (WWW '19)*. 228–239.

[26] Li Chen and Pearl Pu. 2012. Critiquing-based Recommenders: Survey and Emerging Trends. *User Modeling and User-Adapted Interaction* 22, 1-2 (2012), 125–150.

[27] Minmin Chen, Alex Beutel, Paul Covington, Sagar Jain, Francois Belletti, and Ed H. Chi. 2019. Top-K Off-Policy Correction for a REINFORCE Recommender System. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining (WSDM '19)*. 456–464.

[28] Qibin Chen, Junyang Lin, Yichang Zhang, Ming Ding, Yukuo Cen, Hongxia Yang, and Jie Tang. 2019. Towards Knowledge-Based Recommender Dialog System. 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 '2019)*. 1803–1813.

[29] Xinshi Chen, Shuang Li, Hui Li, Shaohua Jiang, Yuan Qi, and Le Song. 2019. Generative adversarial user model for reinforcement learning based recommendation system. In *International Conference on Machine Learning (ICML '19)*. PMLR, 1052–1061.

[30] Zhongxia Chen, Xiting Wang, Xing Xie, Mehul Parsana, Akshay Soni, Xiang Ao, and Enhong Chen. 2020. Towards Explainable Conversational Recommendation. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence, IJCAI '20*. 2994–3000.

[31] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. 2016. Wide & Deep Learning for Recommender Systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (DLRS '2016)*. 7–10.

[32] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP '14)*. 1724–1734.

[33] Ku-Chun Chou, Hsuan-Tien Lin, Chao-Kai Chiang, and Chi-Jen Lu. 2015. Pseudo-reward Algorithms for Contextual Bandits with Linear Payoff Functions. In *Asian Conference on Machine Learning (ACML '15)*. 344–359.

[34] Konstantina Christakopoulou, Alex Beutel, Rui Li, Sagar Jain, and Ed H. Chi. 2018. Q&R: A Two-Stage Approach toward Interactive Recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18)*. 139–148.

[35] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards Conversational Recommender Systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*. 815–824.

[36] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep Neural Networks for YouTube Recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16)*. 191–198.

[37] Chen Cui, Wenjie Wang, Xuemeng Song, Minlie Huang, Xin-Shun Xu, and Liqiang Nie. 2019. User Attention-Guided Multimodal Dialog Systems. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'19)*. 445–454.

[38] Mostafa Dehghani, Sascha Rothe, Enrique Alfonseca, and Pascal Fleury. 2017. Learning to Attend, Copy, and Generate for Session-Based Query Suggestion. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM '17)*. 1747–1756.

[39] Li Deng, Gokhan Tur, Xiaodong He, and Dilek Hakkani-Tur. 2012. Use of Kernel Deep Convex Networks and End-To-End Learning for Spoken Language Understanding. In *2012 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 210–215.

[40] Anoop Deoras and Ruhi Sarikaya. 2013. Deep Belief Network based Semantic Taggers for Spoken Language Understanding. In *ISCA Interspeech*. 2713–2717.

[41] Jan Deriu, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echegoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak. 2021. Survey on Evaluation Methods for Dialogue Systems. *Artificial Intelligence Review* 54, 1 (2021), 755–810.

[42] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (NAACL '19)*. 4171–4186.

[43] Jingtao Ding, Yuhan Quan, Xiangnan He, Yong Li, and Depeng Jin. 2019. Reinforced Negative Sampling for Recommendation with Exposure Data. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. 2230–2236.

[44] Qinxu Ding, Yong Liu, Chunyan Miao, Fei Cheng, and Haihong Tang. 2020. A Hybrid Bandit Framework for Diversified Recommendation. *Proceedings of the AAAI Conference on Artificial Intelligence* (2020).

[45] Jesse Dodge, Andreea Gane, Xiang Zhang, Antoine Bordes, Sumit Chopra, Alexander H. Miller, Arthur Szlam, and Jason Weston. 2016. Evaluating Prerequisite Qualities for Learning End-to-End Dialog Systems. In *International Conference on Learning Representations (ICLR '16)*.

[46] Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to Ask: Neural Question Generation for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL '17)*. 1342–1352.

[47] Yan Duan, John Schulman, Xi Chen, Peter L Bartlett, Ilya Sutskever, and Pieter Abbeel. 2016. Rl ^{2}: Fast Reinforcement Learning via Slow Reinforcement Learning. *arXiv preprint arXiv:1611.02779* (2016).

[48] Travis Ebesu, Bin Shen, and Yi Fang. 2018. Collaborative Memory Network for Recommendation Systems. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '18)*. 515–524.

[49] Minwei Feng, Bing Xiang, Michael R Glass, Lidan Wang, and Bowen Zhou. 2015. Applying deep learning to answer selection: A study and an open task. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 813–820.

[50] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML '17, Vol. 70)*. 1126–1135.

[51] Zuohui Fu, Yikun Xian, Yaxin Zhu, Shuyuan Xu, Zelong Li, Gerard de Melo, and Yongfeng Zhang. 2021. HOOPS: Human-in-the-Loop Graph Reasoning for Conversational Recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*. ACM.

[52] Zuohui Fu, Yikun Xian, Yaxin Zhu, Yongfeng Zhang, and Gerard de Melo. 2020. COOKIE: A Dataset for Conversational Recommendation over Knowledge Graphs in E-commerce. *arXiv preprint arXiv:2008.09237* (2020).

[53] Sudeep Gandhe and David Traum. 2008. An Evaluation Understudy for Dialogue Coherence Models. In *Proceedings of the 9th SIGDIAL Workshop on Discourse and Dialogue (SIGDIAL '08)*. 172–181.

[54] Chongming Gao, Shuai Yuan, Zhong Zhang, Hongzhi Yin, and Junming Shao. 2019. BLOMA: Explain Collaborative Filtering via Boosted Local Rank-One Matrix Approximation. In *International Conference on Database Systems for Advanced Applications (DASFAA '19)*. Springer, 487–490.

[55] Jianfeng Gao, Michel Galley, and Lihong Li. 2019. *Neural Approaches to Conversational AI: Question Answering, Task-oriented Dialogues and Social Chatbots*. Now Foundations and Trends.

[56] Shen Gao, Xiuying Chen, Zhaochun Ren, Dongyan Zhao, and Rui Yan. 2020. Meaningful Answer Generation of E-Commerce Question-Answering. *arXiv preprint arXiv:2011.07307* (2020).

[57] Xiang Gao, Sungjin Lee, Yizhe Zhang, Chris Brockett, Michel Galley, Jianfeng Gao, and Bill Dolan. 2019. Jointly Optimizing Diversity and Relevance in Neural Response Generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL '19)*. 1229–1238.

[58] Asma Ghandeharioun, Judy Hanwen Shen, Natasha Jaques, Craig Ferguson, Noah Jones, Agata Lapedriza, and Rosalind Picard. 2019. Approximating Interactive Human Evaluation with Self-Play for Open-Domain Dialog Systems. In *Advances in Neural Information Processing Systems (NeurIPS '19, Vol. 32)*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.).

[59] Alexandre Gilotte, Clément Calauzènes, Thomas Nedelec, Alexandre Abraham, and Simon Dollé. 2018. Offline A/B Testing for Recommender Systems. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM '18)*. 198–206.

[60] Mark P. Graus and Martijn C. Willemsen. 2015. Improving the User Experience during Cold Start through Choice-Based Preference Elicitation. In *Proceedings of the 9th ACM Conference on Recommender Systems (RecSys '15)*. 273–276.

[61] Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. 2020. A Survey on Knowledge Graph-based Recommender Systems. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[62] Shengbo Guo and Scott Sanner. 2010. Real-time Multiattribute Bayesian Preference Elicitation with Pairwise Comparison Queries. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS '10)*. 289–296.

[63] Javeria Habib, Shuo Zhang, and Krisztian Balog. 2020. IAI MovieBot: A Conversational Movie Recommender System. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20)*. 3405–3408.

[64] Shirley Anugrah Hayati, Dongyeop Kang, Qingxiaoyang Zhu, Weiyan Shi, and Zhou Yu. 2020. INSPIRED: Toward Sociable Recommendation Dialog Systems. In *Conference on Empirical Methods in Natural Language Processing (EMNLP '20)*.

[65] Chen He, Denis Parra, and Katrien Verbert. 2016. Interactive Recommender Systems: A Survey of the State of the Art and Future Research Challenges and Opportunities. *Expert Systems with Applications* 56 (2016), 9 – 27.

[66] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, YongDong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*. 639–648.

[67] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web (WWW '17)*. 173–182.

[68] José Miguel Hernández-Lobato, Neil Houlsby, and Zoubin Ghahramani. 2014. Probabilistic Matrix Factorization with Non-Random Missing Data. In *Proceedings of the 31st International Conference on International Conference on Machine Learning (ICML'14)*. II–1512–II–1520.

[69] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[70] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *Advances in neural information processing systems*. 2042–2050.

[71] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S. Yu. 2018. Leveraging Meta-Path Based Context for Top-N Recommendation with A Neural Co-Attention Model. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18)*. 1531–1540.

[72] Yujing Hu, Qing Da, Anxiang Zeng, Yang Yu, and Yinghui Xu. 2018. Reinforcement Learning to Rank in E-Commerce Search Engine: Formalization, Analysis, and Application. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18)*. 368–377.

[73] Jin Huang, Harrie Oosterhuis, Maarten de Rijke, and Herke van Hoof. 2020. Keeping Dataset Biases out of the Simulation: A Debiased Simulator for Reinforcement Learning Based Recommender Systems. In *Fourteenth ACM Conference on Recommender Systems (RecSys '20)*. 190–199.

[74] Liang Huang, Kai Zhao, and Mingbo Ma. 2017. When to Finish? Optimal Beam Search for Neural Text Generation (modulo beam size). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP '17)*. 2134–2139.

[75] Eugene Ie, Vihan Jain, Jing Wang, Sanmit Narvekar, Ritesh Agarwal, Rui Wu, Heng-Tze Cheng, Morgane Lustman, Vince Gatto, Paul Covington, et al. 2019. Reinforcement learning for slate-based recommender systems: A tractable decomposition and practical methodology. *arXiv preprint arXiv:1905.12767* (2019).

[76] Andrea Iovine, Fedelucio Narducci, and Marco de Gemmis. 2019. A Dataset of Real Dialogues for Conversational Recommender Systems.. In *CLiC-it*.

[77] Andrea Iovine, Fedelucio Narducci, and Giovanni Semeraro. 2020. Conversational Recommender Systems and Natural Language: A Study through the ConveRSE Framework. *Decision Support Systems* 131 (2020), 113250.

[78] Daphne Ippolito, Reno Kriz, João Sedoc, Maria Kustikova, and Chris Callison-Burch. 2019. Comparison of Diverse Decoding Methods from Conditional Language Models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL '19)*. 3752–3762.

[79] Rolf Jagerman, Krisztian Balog, and Maarten de Rijke. 2018. Opensearch: Lessons Learned From an Online Evaluation Campaign. *Journal of Data and Information Quality (JDIQ)* 10, 3 (2018), 1–15.

[80] Rolf Jagerman, Ilya Markov, and Maarten de Rijke. 2019. When People Change Their Mind: Off-Policy Evaluation in Non-Stationary Recommendation Environments. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining (WSDM '19)*. 447–455.

[81] Dietmar Jannach and Ahtsham Manzoor. 2020. End-to-End Learning for Conversational Recommendation: A Long Way to Go? *Proceedings of the 7th Joint Workshop on Interfaces and Human Decision Making for Recommender Systems co-located with 14th ACM Conference on Recommender Systems (RecSys 2020)* (2020).

[82] Dietmar Jannach, Ahtsham Manzoor, Wanling Cai, and Li'e Chen. 2020. A Survey on Conversational Recommender Systems. *arXiv* abs/2004.00646 (2020).

[83] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated Gain-Based Evaluation of IR Techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (Oct. 2002), 422–446.

[84] Hai Jiang, Xin Qi, and He Sun. 2014. Choice-based Recommender Systems: A Unified Approach to Achieving Relevancy and Diversity. *Operations Research* 62, 5 (2014), 973–993.

[85] Xisen Jin, Wenqiang Lei, Zhaochun Ren, Hongshen Chen, Shangsong Liang, Yihong Zhao, and Dawei Yin. 2018. Explicit State Tracking with Semi-Supervisionfor Neural Dialogue Generation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM '18)*. 1403–1412.

[86] Dongyeop Kang, Anusha Balakrishnan, Pararth Shah, Paul Crook, Y-Lan Boureau, and Jason Weston. 2019. Recommendation as a Communication Game: Self-Supervised Bot-Play for Goal-oriented Dialogue. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP '19)*. 1951–1961.

[87] Michael N. Katehakis and Arthur F. Veinott. 1987. The Multi-Armed Bandit Problem: Decomposition and Computation. *Mathematics of Operations Research* 12, 2 (May 1987), 262–268.

[88] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS '17)*. 3149–3157.

[89] Pei Ke, Jian Guan, Minlie Huang, and Xiaoyan Zhu. 2018. Generating Informative Responses with Controlled Sentence Function. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL '18)*. 1499–1508.

[90] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP '14)*. 1746–1751.

[91] Thomas N Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. (2017).

[92] Walid Krichene and Steffen Rendle. 2020. On Sampled Metrics for Item Recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '20)*. 1748–1757.

[93] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask Me Anything: Dynamic Memory Networks for Natural Language Processing. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning (ICML '16)*. 1378–1387.

[94] Branislav Kveton, Csaba Szepesvari, Zheng Wen, and Azin Ashkan. 2015. Cascading Bandits: Learning to Rank in the Cascade Model. In *International Conference on Machine Learning (ICML '15)*. 767–776.

[95] Mirella Lapata. 2003. Probabilistic Text Structuring: Experiments with Sentence Ordering. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1 (ACL '03)*. 545–552.

[96] Mirella Lapata and Regina Barzilay. 2005. Automatic Evaluation of Text Coherence: Models and Representations. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI '05)*. 1085–1090.

[97] Hoyeop Lee, Jinbae Im, Seongwon Jang, Hyunsouk Cho, and Sehee Chung. 2019. MeLU: Meta-Learned User Preference Estimator for Cold-Start Recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19)*. 1073–1082.

[98] Wenqiang Lei, Xiangnan He, Yisong Miao, Qingyun Wu, Richang Hong, Min-Yen Kan, and Tat-Seng Chua. 2020. Estimation-Action-Reflection: Towards Deep Interaction Between Conversational and Recommender Systems. In *Proceedings of the 13th International Conference on Web Search and Data Mining (WSDM' 20)*. ACM, 304–312.

[99] 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 (ACL '18)*. 1437–1447.

[100] Wenqiang Lei, Gangyi Zhang, Xiangnan He, Yisong Miao, Xiang Wang, Liang Chen, and Tat-Seng Chua. 2020. Interactive Path Reasoning on Graph for Conversational Recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '20)*. 2073–2083.

[101] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. 2020. Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems. *arXiv preprint arXiv:2005.01643* (2020).

[102] Mike Lewis, Denis Yarats, Yann Dauphin, Devi Parikh, and Dhruv Batra. 2017. Deal or No Deal? End-to-End Learning of Negotiation Dialogues. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP '17)*. 2443–2453.

[103] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A Diversity-Promoting Objective Function for Neural Conversation Models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL '16)*. 110–119.

[104] Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016. Deep Reinforcement Learning for Dialogue Generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP '16)*. 1192–1202.

[105] Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. 2010. A Contextual-Bandit Approach to Personalized News Article Recommendation. In *Proceedings of the 19th International Conference on World Wide Web (WWW '10)*. 661–670.

[106] Lihong Li, Jin Young Kim, and Imed Zitouni. 2015. Toward Predicting the Outcome of an A/B Experiment for Search Relevance. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining (WSDM '15)*. 37–46.

[107] Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards Deep Conversational Recommendations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NeurIPS '18)*. 9748–9758.

[108] Shijun Li, Wenqiang Lei, Qingyun Wu, Xiangnan He, Peng Jiang, and Tat-Seng Chua. 2021. Seamlessly Unifying Attributes and Items: Conversational Recommendation for Cold-Start Users. *ACM Transactions on Information Systems* (2021).

[109] Ziming Li, Julia Kiseleva, and Maarten de Rijke. 2021. Improving Response Quality with Backward-reasoning in Open-domain Dialogue Systems. In *SIGIR 2021: 44th international ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM.

[110] Defu Lian, Qi Liu, and Enhong Chen. 2020. Personalized Ranking with Importance Sampling. In *Proceedings of The Web Conference 2020 (WWW '20)*. 1093–1103.

[111] Lizi Liao, Yunshan Ma, Xiangnan He, Richang Hong, and Tat-Seng Chua. 2018. Knowledge-Aware Multimodal Dialogue Systems. In *Proceedings of the 26th ACM International Conference on Multimedia (MM '18)*. 801–809.

[112] Lizi Liao, Ryuichi Takanobu, Yunshan Ma, Xun Yang, Minlie Huang, and Tat-Seng Chua. 2020. Topic-Guided Relational Conversational Recommender in Multi-Domain. *IEEE Transactions on Knowledge and Data Engineering (TKDE)* (2020).

[113] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*. 74–81.

[114] 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 *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP '16)*. 2122–2132.

[115] Dugang Liu, Pengxiang Cheng, Zhenhua Dong, Xiuqiang He, Weike Pan, and Zhong Ming. 2020. A General Knowledge Distillation Framework for Counterfactual Recommendation via Uniform Data. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*. 831–840.

[116] Weiwen Liu, Qing Liu, Ruiming Tang, Junyang Chen, Xiuqiang He, and Pheng Ann Heng. 2020. Personalized Re-Ranking with Item Relationships for E-Commerce. In *Proceedings of the 29th ACM In-*

ternational Conference on Information & Knowledge Management (CIKM '20). 925–934.

[117] Yiming Liu, Xuezhi Cao, and Yong Yu. 2016. Are You Influenced by Others When Rating? Improve Rating Prediction by Conformity Modeling. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16)*. 269–272.

[118] Yong Liu, Yingtai Xiao, Qiong Wu, Chunyan Miao, Juyong Zhang, Binqiang Zhao, and Haihong Tang. 2020. Diversified Interactive Recommendation with Implicit Feedback. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI '20)*. 4932–4939.

[119] Zeming Liu, Haifeng Wang, Zheng-Yu Niu, Hua Wu, Wanxiang Che, and Ting Liu. 2020. Towards Conversational Recommendation over Multi-Type Dialogs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL '20)*. 1036–1049.

[120] Benedikt Loepp, Tim Hussein, and Jüergen Ziegler. 2014. Choice-Based Preference Elicitation for Collaborative Filtering Recommender Systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. 3085–3094.

[121] Jordan J Louviere, David A Hensher, and Joffre D Swait. 2000. *Stated Choice Methods: Analysis and Applications*. Cambridge University Press.

[122] Zhengdong Lu and Hang Li. 2013. A Deep Architecture for Matching Short Texts. In *Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS '13)*. 1367–1375.

[123] Kai Luo, Scott Sanner, Ga Wu, Hanze Li, and Hojin Yang. 2020. Latent Linear Critiquing for Conversational Recommender Systems. In *Proceedings of The Web Conference 2020 (WWW' 20)*. 2535–2541.

[124] Kai Luo, Hojin Yang, Ga Wu, and Scott Sanner. 2020. Deep Critiquing for VAE-Based Recommender Systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*. 1269–1278.

[125] Hao Ma, Haixuan Yang, Irwin King, and Michael R. Lyu. 2008. Learning Latent Semantic Relations from Clickthrough Data for Query Suggestion. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM '08)*. 709–718.

[126] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. 2019. Learning Disentangled Representations for Recommendation. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS '20)*. 5711–5722.

[127] Wenchang Ma, Ryuichi Takanobu, Minghao Tu, and Minlie Huang. 2020. Bridging the Gap between Conversational Reasoning and Interactive Recommendation. *arXiv preprint arXiv:2010.10333* (2020).

[128] Zhengyi Ma, Zhicheng Dou, Yutao Zhu, Hanxun Zhong, and Ji-Rong Wen. 2021. One Chatbot Per Person: Creating Personalized Chatbots based on Implicit User Profiles. In *SIGIR 2021: 44th international ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*.

[129] Tariq Mahmood and Francesco Ricci. 2007. Learning and Adaptivity in Interactive Recommender Systems. In *Proceedings of the Ninth International Conference on Electronic Commerce (ICEC '07)*. 75–84.

[130] Francesca Mangili, Denis Broggini, Alessandro Antonucci, Marco Alberti, and Lorenzo Cimasoni. 2020. A Bayesian Approach to Conversational Recommendation Systems. *AAAI 2020 Workshop on Interactive and Conversational Recommendation Systems* (2020).

[131] Benjamin M. Marlin, Richard S. Zemel, Sam Roweis, and Malcolm Slaney. 2007. Collaborative Filtering and the Missing at Random Assumption. In *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence (UAI '07)*. 267–275.

[132] Julian McAuley, Rahul Pandey, and Jure Leskovec. 2015. Inferring Networks of Substitutable and Complementary Products. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '15)*. 785–794.

[133] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-Based Recommendations on Styles and Substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '15)*. 43–52.

[134] Kevin McCarthy, James Reilly, Lorraine McGinty, and Barry Smyth. 2004. Thinking positively-explanatory feedback for conversational recommender systems. In *Proceedings of the European Conference on Case-Based Reasoning (ECCBR-04) Explanation Workshop*. 115–124.

[135] James McInerney, Brian Brost, Praveen Chandar, Rishabh Mehrotra, and Benjamin Carterette. 2020. Counterfactual Evaluation of Slate Recommendations with Sequential Reward Interactions. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '20)*. 1779–1788.

[136] Grégoire Mesnil, Xiaodong He, Li Deng, and Yoshua Bengio. 2013. Investigation of Recurrent-Neural-Network Architectures and Learning Methods for Spoken Language Understanding. In *Interspeech*. 3771–3775.

[137] Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-Value Memory Networks for Directly Reading Documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP '16)*. 1400–1409.

[138] Nader Mirzadeh, Francesco Ricci, and Mukesh Bansal. 2005. Feature Selection Methods for Conversational Recommender Systems. In *Proceedings of the 2005 IEEE International Conference on E-Technology, e-Commerce and e-Service (EEE'05) on e-Technology, e-Commerce and e-Service (EEE '05)*. 772–777.

[139] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing Atari with Deep Reinforcement Learning. *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS) Deep Learning Workshop* (2013).

[140] Kaixiang Mo, Yu Zhang, Shuangyin Li, Jiajun Li, and Qiang Yang. 2018. Personalizing a Dialogue System With Transfer Reinforcement Learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI '18)*.

[141] Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. 2019. OpenDialKG: Explainable Conversational Reasoning with Attention-based Walks over Knowledge Graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL '19)*. 845–854.

[142] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Ranking Sentences for Extractive Summarization with Reinforcement Learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL '18)*. 1747–1759.

[143] John Ashworth Nelder and Robert WM Wedderburn. 1972. Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)* 135, 3 (1972), 370–384.

[144] Andrew Y. Ng and Stuart J. Russell. 2000. Algorithms for Inverse Reinforcement Learning. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML '00)*. 663–670.

[145] Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017. Why We Need New Evaluation Metrics for NLG. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP '16)*. 2241–2252.

[146] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text Matching as Image Recognition. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI '16)*. 2793–2799.

[147] 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.

[148] 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 (ACL '02). 311–318.

[149] Florian Pecune, Lucile Callebert, and Stacy Marsella. 2020. A Socially-Aware Conversational Recommender System for Personalized Recipe Recommendations. In *Proceedings of the 8th International Conference on Human-Agent Interaction (HAI '20)*. 78–86.

[150] Florian Pecune, Shruti Murali, Vivian Tsai, Yoichi Matsuyama, and Justine Cassell. 2019. A Model of Social Explanations for a Conversational Movie Recommendation System. In *Proceedings of the 7th International Conference on Human-Agent Interaction (HAI '19)*. 135–143.

[151] Jiahuan Pei, Pengjie Ren, and Maarten de Rijke. 2021. A Cooperative Memory Network for Personalized Task-oriented Dialogue Systems with Incomplete User Profiles. In *The Web Conference 2021*. ACM.

[152] Gustavo Penha and Claudia Hauff. 2020. What Does BERT Know about Books, Movies and Music? Probing BERT for Conversational Recommendation. In *Fourteenth ACM Conference on Recommender Systems (RecSys '20)*. 388–397.

[153] Johannes Pittermann, Angela Pittermann, and Wolfgang Minker. 2010. Emotion recognition and adaptation in spoken dialogue systems. *International Journal of Speech Technology* 13, 1 (2010), 49–60.

[154] Bruno Pradel, Nicolas Usunier, and Patrick Gallinari. 2012. Ranking with Non-Random Missing Ratings: Influence of Popularity and Positivity on Evaluation Metrics. In *Proceedings of the Sixth ACM Conference on Recommender Systems (RecSys '12)*. 147–154.

[155] Pearl Pu and Boi Faltings. 2004. Decision tradeoff using example-critiquing and constraint programming. *Constraints* 9, 4 (2004), 289–310.

[156] Qiao Qian, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Assigning Personality/Profile to a Chatting Machine for Coherent Conversation Generation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI '18)*. 4279–4285.

[157] Lijing Qin, Shouyuan Chen, and Xiaoyan Zhu. 2014. Contextual Combinatorial Bandit and its Application on Diversified Online Recommendation. In *Proceedings of the 2014 SIAM International Conference on Data Mining (SDM '14)*. SIAM, 461–469.

[158] Lisong Qiu, Juntao Li, Wei Bi, Dongyan Zhao, and Rui Yan. 2019. Are Training Samples Correlated? Learning to Generate Dialogue Responses with Multiple References. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL '19)*. 3826–3835.

[159] Xipeng Qiu and Xuanjing Huang. 2015. Convolutional Neural Tensor Network Architecture for Community-Based Question Answering. In *Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI '15)*. 1305–1311.

[160] Arpit Rana and Derek Bridge. 2020. Navigation-by-Preference: A New Conversational Recommender with Preference-Based Feedback. In *Proceedings of the 25th International Conference on Intelligent User Interfaces (IUI '20)*. 155–165.

[161] Pengjie Ren, Zhongkun Liu, Xiaomeng Song, Hongtao Tian, Zhumin Chen, Zhaochun Ren, and Maarten de Rijke. 2021. Wizard of Search Engine: Access to Information Through Conversations with Search Engines. In *SIGIR 2021: 44th international ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM.

[162] Xuhui Ren, Hongzhi Yin, Tong Chen, Hao Wang, Quoc Viet Hung Nguyen, Zi Huang, and Xiangliang Zhang. 2020. CRSAL: Conversational Recommender Systems with Adversarial Learning. *ACM Transactions on Information Systems* 0, ja (2020), 1.

[163] Steffen Rendle. 2010. Factorization Machines. In *Proceedings of the 2010 IEEE International Conference on Data Mining (ICDM '10)*. 995–1000.

[164] Corbin Rosset, Chenyan Xiong, Xia Song, Daniel Campos, Nick Craswell, Saurabh Tiwary, and Paul Bennett. 2020. Leading Conversational Search by Suggesting Useful Questions. In *Proceedings of The Web Conference 2020 (WWW '20)*. 1160–1170.

[165] Paula Saavedra, Pablo Barreiro, Roi Duran, Rosa Crujeiras, María Loureiro, and Eduardo Sánchez Vila. 2016. Choice-Based Recommender Systems. In *ACM RecSys Workshop on Recommenders in Tourism*. 38–46.

[166] Otmane Sakhi, Stephen Bonner, David Rohde, and Flavian Vasile. 2020. BLOB: A Probabilistic Model for Recommendation that Combines Organic and Bandit Signals. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '20)*. 783–793.

[167] Ruslan Salakhutdinov and Andriy Mnih. 2007. Probabilistic Matrix Factorization. In *Proceedings of the 20th International Conference on Neural Information Processing Systems (NIPS '07)*. 1257–1264.

[168] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-Based Collaborative Filtering Recommendation Algorithms. In *Proceedings of the 10th International Conference on World Wide Web (WWW '01)*. 285–295.

[169] J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. 2007. Collaborative filtering recommender systems. In *The adaptive web*. Springer, 291–324.

[170] Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. 2007. Agenda-Based User Simulation for Bootstrapping a POMDP Dialogue System. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers (NAACL-Short '07)*. 149–152.

[171] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling Relational Data with Graph Convolutional Networks. In *European Semantic Web Conference (ESWC '18)*. Springer, 593–607.

[172] Tobias Schnabel, Paul N. Bennett, Susan T. Dumais, and Thorsten Joachims. 2018. Short-Term Satisfaction and Long-Term Coverage: Understanding How Users Tolerate Algorithmic Exploration. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM '18)*. 513–521.

[173] John Schulman, Sergey Levine, Philipp Moritz, Michael Jordan, and Pieter Abbeel. 2015. Trust Region Policy Optimization. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning (ICML '15)*. 1889–1897.

[174] Anna Sepliarskaia, Julia Kiseleva, Filip Radlinski, and Maarten de Rijke. 2018. Preference Elicitation as an Optimization Problem. In *Proceedings of the 12th ACM Conference on Recommender Systems (RecSys '18)*. 172–180.

[175] Ashish Sharma, Inna W Lin, Adam S Miner, David C Atkins, and Tim Althoff. 2021. Towards Facilitating Empathic Conversations in Online Mental Health Support: A Reinforcement Learning Approach. In *Proceedings of The Web Conference 2021 (WWW '21)*.

[176] Yue Shi, Martha Larson, and Alan Hanjalic. 2014. Collaborative Filtering beyond the User-Item Matrix: A Survey of the State of the Art and Future Challenges. *Comput. Surveys* 47, 1, Article 3 (May 2014), 45 pages.

[177] Thiago Silveira, Min Zhang, Xiao Lin, Yiqun Liu, and Shaoping Ma. 2019. How Good Your Recommender System Is? A Survey on Evaluations in Recommendation. *International Journal of Machine Learning and Cybernetics* 10, 5 (2019), 813–831.

[178] Ayan Sinha, David F. Gleich, and Karthik Ramani. 2016. Deconvolving Feedback Loops in Recommender Systems. In *Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS '16)*. 3251–3259.

[179] Barry Smyth and Lorraine McGinty. 2003. An Analysis of Feedback Strategies in Conversational Recommenders. In *the Fourteenth Irish Artificial Intelligence and Cognitive Science Conference (AICS '2003)*. Citeseer.

[180] Barry Smyth, Lorraine McGinty, James Reilly, and Kevin McCarthy. 2004. Compound Critiques for Conversational Recommender Systems. In *Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence (WI '04)*. 145–151.

[181] Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In

Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI '17). 4444–4451.

[182] Harald Steck. 2011. Item Popularity and Recommendation Accuracy. In *Proceedings of the Fifth ACM Conference on Recommender Systems (RecSys '11)*. 125–132.

[183] Harald Steck. 2013. Evaluation of Recommendations: Rating-Prediction and Ranking. In *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys '13)*. 213–220.

[184] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-End Memory Networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS '15)*. 2440–2448.

[185] Wenlong Sun, Sami Khenissi, Olfa Nasraoui, and Patrick Shafto. 2019. Debiasing the Human-Recommender System Feedback Loop in Collaborative Filtering. In *Companion Proceedings of The 2019 World Wide Web Conference (WWW '19)*. 645–651.

[186] Weiwei Sun, Shuo Zhang, Krisztian Balog, Zhaochun Ren, Pengjie Ren, Zhumin Chen, and Maarten de Rijke. 2021. Simulating User Satisfaction for the Evaluation of Task-oriented Dialogue Systems. In *SIGIR 2021: 44th international ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM.

[187] Yueming Sun and Yi Zhang. 2018. Conversational Recommender System. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '18)*. 235–244.

[188] Peter Sunehag, Richard Evans, Gabriel Dulac-Arnold, Yori Zwols, Daniel Visentin, and Ben Coppin. 2015. Deep Reinforcement Learning with Attention for Slate Markov Decision Processes with High-dimensional States and Actions. *arXiv preprint arXiv:1512.01124* (2015).

[189] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2 (NIPS '14)*. 3104–3112.

[190] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*.

[191] Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2016. Improved Representation Learning for Question Answer Matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL '16)*. 464–473.

[192] Cynthia A. Thompson, Mehmet H. Göker, and Pat Langley. 2004. A Personalized System for Conversational Recommendations. *Journal of Artificial Intelligence Research* 21, 1 (March 2004), 393–428.

[193] Frederich N Tou, Michael D Williams, Richard Fikes, D Austin Henderson Jr, and Thomas W Malone. 1982. RABBIT: An Intelligent Database Assistant. In *Proceedings of the National Conference on Artificial Intelligence (AAAI '82)*. 314–318.

[194] Daisuke Tsumita and Tomohiro Takagi. 2019. Dialogue Based Recommender System That Flexibly Mixes Utterances and Recommendations. In *IEEE/WIC/ACM International Conference on Web Intelligence (WI '19)*. 51–58.

[195] Amos Tversky and Itamar Simonson. 1993. Context-Dependent Preferences. *Management Science* 39, 10 (Oct. 1993), 1179–1189.

[196] Svitlana Vakulenko, Evangelos Kanoulas, and Maarten de Rijke. 2020. An Analysis of Mixed Initiative and Collaboration in Information-Seeking Dialogues. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*. 2085–2088.

[197] Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2019. Deep Graph Infomax. In *International Conference on Learning Representations (ICLR '19)*.

[198] Ivan Vendrov, Tyler Lu, Qingqing Huang, and Craig Boutilier. 2020. Gradient-based Optimization for Bayesian Preference Elicitation. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI '20)*.

[199] Paolo Viappiani, Pearl Pu, and Boi Faltings. 2007. Conversational Recommenders with Adaptive Suggestions. In *Proceedings of the 2007 ACM Conference on Recommender Systems (RecSys '07)*. 89–96.

[200] Nikos Voskarides, Dan Li, Pengjie Ren, Evangelos Kanoulas, and Maarten de Rijke. 2020. Query Resolution for Conversational Search with Limited Supervision. In *SIGIR 2020: 43rd international ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 921–932.

[201] Mengting Wan, Di Wang, Jie Liu, Paul Bennett, and Julian McAuley. 2018. Representing and Recommending Shopping Baskets with Complementarity, Compatibility and Loyalty. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM '18)*. 1133–1142.

[202] Shengxian Wan, Yanyan Lan, Jiafeng Guo, Jun Xu, Liang Pang, and Xueqi Cheng. 2016. A Deep Architecture for Semantic Matching with Multiple Positional Sentence Representations. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI '16)*. 2835–2841.

[203] Bingning Wang, Kang Liu, and Jun Zhao. 2016. Inner Attention based Recurrent Neural Networks for Answer Selection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL '16)*. 1288–1297.

[204] Chaoyang Wang, Zhiqiang Guo, Jianjun Li, Peng Pan, and Guohui Li. 2020. A Text-Based Deep Reinforcement Learning Framework for Interactive Recommendation. In *Proceedings of the 24th European Conference on Artificial Intelligence (ECAI '20, Vol. 325)*. 537–544.

[205] Huazheng Wang, Qingyun Wu, and Hongning Wang. 2017. Factorization Bandits for Interactive Recommendation. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI '17)*. 2695–2702.

[206] Hao-nan Wang, Ning Liu, Yi-yun Zhang, Da-wei Feng, Feng Huang, Dong-sheng Li, and Yi-ming Zhang. 2020. Deep Reinforcement Learning: A Survey. *Frontiers of Information Technology & Electronic Engineering* (2020), 1–19.

[207] Jane X Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Remi Munos, Charles Blundell, Dharshan Kumaran, and Matt Botvinick. 2016. Learning to Reinforcement Learn. *arXiv preprint arXiv:1611.05763* (2016).

[208] Qing Wang, Chunqiu Zeng, Wubai Zhou, Tao Li, S Sitharama Iyengar, Larisa Shwartz, and Genady Ya Grabarnik. 2018. Online Interactive Collaborative Filtering Using Multi-Armed Bandit with Dependent Arms. *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 31, 8 (2018), 1569–1580.

[209] Shuohang Wang and Jing Jiang. 2016. Learning Natural Language Inference with LSTM. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL '16)*. 1442–1451.

[210] Weiquan Wang and Izak Benbasat. 2013. Research Note—A Contingency Approach to Investigating the Effects of User-system Interaction Modes of Online Decision Aids. *Information Systems Research* 24, 3 (2013), 861–876.

[211] Wenjie Wang, Fuli Feng, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2020. Denoising Implicit Feedback for Recommendation. *arXiv preprint arXiv:2006.04153* (2020).

[212] Wenjie Wang, Fuli Feng, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. 2020. "Click" Is Not Equal to "Like": Counterfactual Recommendation for Mitigating Clickbait Issue. *arXiv preprint arXiv:2009.09945* (2020).

[213] Xin Wang, Steven C.H. Hoi, Chenghao Liu, and Martin Ester. 2017. Interactive Social Recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM '17)*. 357–366.

[214] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled Graph Collaborative Filtering. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*.

[215] Xuewei Wang, Weiyan Shi, Richard Kim, Yoojung Oh, Sijia Yang, Jingwen Zhang, and Zhou Yu. 2019. Persuasion for Good: Towards a Personalized Persuasive Dialogue System for Social Good. In *Pro-*

ceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL '19). 5635–5649.

[216] Xiang Wang, Yaokun Xu, Xiangnan He, Yixin Cao, Meng Wang, and Tat-Seng Chua. 2020. Reinforced Negative Sampling over Knowledge Graph for Recommendation. In *Proceedings of The Web Conference 2020 (WWW '20)*. 99–109.

[217] Tianxin Wei, Ziwei Wu, Ruirui Li, Ziniu Hu, Fuli Feng, Xiangnan He, Yizhou Sun, and Wei Wang. 2020. Fast Adaptation for Cold-Start Collaborative Filtering with Meta-Learning. In *2019 IEEE International Conference on Data Mining (ICDM '19)*.

[218] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying Graph Convolutional Networks. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*. 6861–6871.

[219] Ga Wu, Kai Luo, Scott Sanner, and Harold Soh. 2019. Deep Language-Based Critiquing for Recommender Systems. In *Proceedings of the 13th ACM Conference on Recommender Systems (RecSys '19)*. 137–145.

[220] Le Wu, Xiangnan He, Xiang Wang, Kun Zhang, and Meng Wang. 2021. A Survey on Neural Recommendation: From Collaborative Filtering to Content and Context Enriched Recommendation. *arXiv preprint arXiv:2104.13030* (2021).

[221] Shiwen Wu, Fei Sun, Wentao Zhang, and Bin Cui. 2020. Graph Neural Networks in Recommender Systems: A Survey. *arXiv preprint arXiv:2011.02260* (2020).

[222] Wenquan Wu, Zhen Guo, Xiangyang Zhou, Hua Wu, Xiyuan Zhang, Rongzhong Lian, and Haifeng Wang. 2019. Proactive Human-Machine Conversation with Explicit Conversation Goal. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL '19)*. 3794–3804.

[223] Wei Wu and Rui Yan. 2018. Deep Chit-Chat: Deep Learning for ChatBots. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*.

[224] Yikun Xian, Zuohui Fu, S. Muthukrishnan, Gerard de Melo, and Yongfeng Zhang. 2019. Reinforcement Knowledge Graph Reasoning for Explainable Recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19)*. 285–294.

[225] Hu Xu, Seungwhan Moon, Honglei Liu, Bing Liu, Pararth Shah, Bing Liu, and Philip Yu. 2020. User Memory Reasoning for Conversational Recommendation. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING '20)*. 5288–5308.

[226] Kerui Xu, Jingxuan Yang, Jun Xu, Sheng Gao, Jun Guo, and Ji-Rong Wen. 2021. Adapting User Preference to Online Feedback in Multi-Round Conversational Recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining (WSDM '21)*. 364–372.

[227] Ya Xu, Nanyu Chen, Addrian Fernandez, Omar Sinno, and Anmol Bhasin. 2015. From Infrastructure to Culture: A/B Testing Challenges in Large Scale Social Networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '15)*. 2227–2236.

[228] Rui Yan, Yiping Song, and Hua Wu. 2016. Learning to Respond with Deep Neural Networks for Retrieval-Based Human-Computer Conversation System. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '16)*. 55–64.

[229] Hojin Yang, Scott Sanner, Ga Wu, and Jin Peng Zhou. 2021. Bayesian Preference Elicitation with Keyphase-Item Coembeddings for Interactive Recommendation. In *Proceedings of the 29th International Conference on User Modeling, Adaptation, and Personalization (UMAP-21)*.

[230] Longqi Yang, Yin Cui, Yuan Xuan, Chenyang Wang, Serge Belongie, and Deborah Estrin. 2018. Unbiased Offline Recommender Evaluation for Missing-Not-at-Random Implicit Feedback. In *Proceedings of the 12th ACM Conference on Recommender Systems (RecSys '18)*. 279–287.

[231] Mengyue Yang, Qingyang Li, Zhiwei Qin, and Jieping Ye. 2020. Hierarchical Adaptive Contextual Bandits for Resource Constraint based Recommendation. In *Proceedings of The Web Conference 2020 (WWW' 20)*. 292–302.

[232] Kaisheng Yao, Baolin Peng, Yu Zhang, Dong Yu, Geoffrey Zweig, and Yangyang Shi. 2014. Spoken Language Understanding using Long Short-Term Memory Neural Networks. In *2014 IEEE Spoken Language Technology Workshop (SLT Workshop)*. IEEE, 189–194.

[233] Kaisheng Yao, Geoffrey Zweig, Mei-Yuh Hwang, Yangyang Shi, and Dong Yu. 2013. Recurrent Neural Networks for Language Understanding. In *Interspeech*. 2524–2528.

[234] Yi-Ting Yeh and Yun-Nung Chen. 2019. QAInfomax: Learning Robust Question Answering System by Mutual Information Maximization. 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)*. 3370–3375.

[235] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. 2018. Graph Convolutional Neural Networks for Web-Scale Recommender Systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18)*. 974–983.

[236] Tom Young, Erik Cambria, Iti Chaturvedi, Hao Zhou, Subham Biswas, and Minlie Huang. 2018. Augmenting End-to-End Dialogue Systems with Commonsense Knowledge. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI '18)*.

[237] Junliang Yu, Min Gao, Hongzhi Yin, Jundong Li, Chongming Gao, and Qinyong Wang. 2019. Generating Reliable Friends via Adversarial Training to Improve Social Recommendation. In *2019 IEEE International Conference on Data Mining (ICDM '19)*. IEEE, 768–777.

[238] Tong Yu, Yilin Shen, and Hongxia Jin. 2019. A Visual Dialog Augmented Interactive Recommender System. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19)*. 157–165.

[239] Chunqiu Zeng, Qing Wang, Shekoofeh Mokhtari, and Tao Li. 2016. Online Context-Aware Recommendation with Time Varying Multi-Armed Bandit. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '16)*. 2025–2034.

[240] Ruiyi Zhang, Tong Yu, Yilin Shen, Hongxia Jin, Changyou Chen, and Lawrence Carin. 2019. Reward Constrained Interactive Recommendation with Natural Language Feedback. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems (NeurIPS '19)*.

[241] Shuo Zhang and Krisztian Balog. 2020. Evaluating Conversational Recommender Systems via User Simulation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '20)*. 1512–1520.

[242] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep Learning Based Recommender System: A Survey and New Perspectives. *ACM Computing Surveys (CSUR)* 52, 1, Article 5 (2019), 38 pages.

[243] Xiaoying Zhang, Hong Xie, Hang Li, and John C.S. Lui. 2020. Conversational Contextual Bandit: Algorithm and Application. In *Proceedings of The Web Conference (WWW '20)*. 662–672.

[244] Yongfeng Zhang and Xu Chen. 2020. Explainable Recommendation: A Survey and New Perspectives. *Foundations and Trends® in Information Retrieval* 14, 1 (2020), 1–101.

[245] 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 (CIKM '18)*. 177–186.

[246] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceed-*

ings of the 37th international ACM SIGIR conference on Research & development in information retrieval (SIGIR '14). 83–92.

[247] Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020. Task-Oriented Dialog Systems that Consider Multiple Appropriate Responses under the Same Context. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI '20)*.

[248] Yongfeng Zhang, Haochen Zhang, Min Zhang, Yiqun Liu, and Shaoping Ma. 2014. Do Users Rate or Review? Boost Phrase-Level Sentiment Labeling with Review-Level Sentiment Classification. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '14)*. 1027–1030.

[249] Zheng Zhang, Lizi Liao, Minlie Huang, Xiaoyan Zhu, and Tat-Seng Chua. 2019. Neural Multimodal Belief Tracker with Adaptive Attention for Dialogue Systems. In *The World Wide Web Conference (WWW '19)*. 2401–2412.

[250] Zheng Zhang, Ryuichi Takanobu, Minlie Huang, and Xiaoyan Zhu. 2020. Recent Advances and Challenges in Task-oriented Dialog System. *arXiv preprint arXiv:2003.07490* (2020).

[251] Xiangyu Zhao, Long Xia, Zhuoye Ding, Dawei Yin, and Jiliang Tang. 2019. Toward simulating environments in reinforcement learning based recommendations. *arXiv preprint arXiv:1906.11462* (2019).

[252] Xiangyu Zhao, Liang Zhang, Zhuoye Ding, Long Xia, Jiliang Tang, and Dawei Yin. 2018. Recommendations with Negative Feedback via Pairwise Deep Reinforcement Learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18)*. 1040–1048.

[253] Xiaoxue Zhao, Weinan Zhang, and Jun Wang. 2013. Interactive Collaborative Filtering. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management (CIKM '13)*. 1411–1420.

[254] Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. 2018. DRN: A Deep Reinforcement Learning Framework for News Recommendation. In *Proceedings of the 2018 World Wide Web Conference (WWW '18)*. 167–176.

[255] Yinhe Zheng, Rongsheng Zhang, Minlie Huang, and Xiaoxi Mao. 2020. A Pre-Training Based Personalized Dialogue Generation Model with Persona-Sparse Data. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI '20)*. 9693–9700.

[256] Chunyi Zhou, Yuanyuan Jin, Xiaoling Wang, and Yingjie Zhang. 2020. Conversational Music Recommendation based on Bandits. In *2020 IEEE International Conference on Knowledge Graph (ICKG '20)*. IEEE, 41–48.

[257] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep Interest Network for Click-Through Rate Prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18)*. 1059–1068.

[258] Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. 2018. Emotional Chatting Machine: Emotional Conversation Generation with Internal and External Memory. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI '18)*.

[259] Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Commonsense Knowledge Aware Conversation Generation with Graph Attention. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI '18)*. 4623–4629.

[260] Kun Zhou, Xiaolei Wang, Yuanhang Zhou, Chenzhan Shang, Yuan Cheng, Wayne Xin Zhao, Yaliang Li, and Ji-Rong Wen. 2021. CRSLab: An Open-Source Toolkit for Building Conversational Recommender System. *arXiv preprint arXiv:2101.00939* (2021).

[261] Kun Zhou, Wayne Xin Zhao, Shuqing Bian, Yuanhang Zhou, Ji-Rong Wen, and Jingsong Yu. 2020. Improving Conversational Recommender Systems via Knowledge Graph based Semantic Fusion. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (SIGKDD' 20)*. 1006–1014.

[262] Kun Zhou, Wayne Xin Zhao, Hui Wang, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. Leveraging Historical Interaction Data for Improving Conversational Recommender System. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management (CIKM '20)*. 2349–2352.

[263] Kun Zhou, Yuanhang Zhou, Wayne Xin Zhao, Xiaoke Wang, and Ji-Rong Wen. 2020. Towards Topic-Guided Conversational Recommender System. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING '2020)*.

[264] Sijin Zhou, Xinyi Dai, Haokun Chen, Weinan Zhang, Kan Ren, Ruiming Tang, Xiuqiang He, and Yong Yu. 2020. Interactive Recommender System via Knowledge Graph-Enhanced Reinforcement Learning. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR' 20)*. 179–188.

[265] Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. 2021. Retrieving and Reading: A Comprehensive Survey on Open-domain Question Answering. *arXiv preprint arXiv:2101.00774* (2021).

[266] Han Zhu, Xiang Li, Pengye Zhang, Guozheng Li, Jie He, Han Li, and Kun Gai. 2018. Learning Tree-Based Deep Model for Recommender Systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18)*. 1079–1088.

[267] Shi Zong, Hao Ni, Kenny Sung, Nan Rosemary Ke, Zheng Wen, and Branislav Kveton. 2016. Cascading Bandits for Large-Scale Recommendation Problems. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence (UAI '16)*. 835–844.

[268] Jie Zou, Yifan Chen, and Evangelos Kanoulas. 2020. Towards Question-Based Recommender Systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*. 881–890.

[269] Lixin Zou, Long Xia, Zhuoye Ding, Jiaxing Song, Weidong Liu, and Dawei Yin. 2019. Reinforcement Learning to Optimize Long-Term User Engagement in Recommender Systems. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19)*. 2810–2818.

[270] Lixin Zou, Long Xia, Pan Du, Zhuo Zhang, Ting Bai, Weidong Liu, Jian-Yun Nie, and Dawei Yin. 2020. Pseudo Dyna-Q: A Reinforcement Learning Framework for Interactive Recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining (WSDM '20)*. 816–824.

[271] Lixin Zou, Long Xia, Yulong Gu, Xiangyu Zhao, Weidong Liu, Jimmy Xiangji Huang, and Dawei Yin. 2020. Neural Interactive Collaborative Filtering. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*. 749–758.