A Survey on Conversational Recommender Systems

DIETMAR JANNACH and AHTSHAM MANZOOR, University of Klagenfurt
WANLING CAI and LI CHEN, Hong Kong Baptist University

Recommender systems are software applications that help users to find items of interest in situations of information overload. Current research often assumes a one-shot interaction paradigm, where the users' preferences are estimated based on past observed behavior and where the presentation of a ranked list of suggestions is the main, one-directional form of user interaction. Conversational recommender systems (CRS) take a different approach and support a richer set of interactions. These interactions can, for example, help to improve the preference elicitation process or allow the user to ask questions about the recommendations and to give feedback. The interest in CRS has significantly increased in the past few years. This development is mainly due to the significant progress in the area of natural language processing, the emergence of new voice-controlled home assistants, and the increased use of chatbot technology. With this article, we provide a detailed survey of existing approaches to conversational recommendation. We categorize these approaches in various dimensions, e.g., in terms of the supported user intents or the knowledge they use in the background. Moreover, we discuss technological approaches, review how CRS are evaluated, and finally identify a number of gaps that deserve more research in the future.

CCS Concepts: • **Information systems** → **Recommender systems**; • **General and reference** → *Surveys and overviews*; • **Human-centered computing** → *Interactive systems and tools*;

Additional Key Words and Phrases: Conversational recommendation, dialogue systems

ACM Reference format:
Dietmar Jannach, Ahtsham Manzoor, Wanling Cai, and Li Chen. 2021. A Survey on Conversational Recommender Systems. *ACM Comput. Surv.* 54, 5, Article 105 (May 2021), 36 pages.
https://doi.org/10.1145/3453154

1 INTRODUCTION

Recommender systems are among the most visible success stories of AI in practice. Typically, the main task of such systems is to point users to potential items of interest, e.g., in the context of an e-commerce site. Thereby, they not only help users in situations of information overload [126], but they also can significantly contribute to the business success of the service providers [57].

In many of these practical applications, recommending is a *one-shot* interaction process. Typically, the underlying system monitors the behavior of its users over time and then presents a tailored set of recommendations in pre-defined navigational situations, e.g., when a user logs in to the service. Although such an approach is common and useful in various domains, it can have a

Authors' addresses: D. Jannach and A. Manzoor, Department of Artificial Intelligence and Cybersecurity, University of Klagenfurt, Universitätsstraße 65-67, 9020 Klagenfurt, Austria; emails: {dietmar.jannach, ahtsham.manzoor}@aau.at; W. Cai and L. Chen, DLB 643, Level 6, David C. Lam Building, Shaw Campus, Hong Kong Baptist University, Kowloon Tong, Hong Kong; emails: {cswlcai, lichen}@comp.hkbu.edu.hk.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.

0360-0300/2021/05-ART105 $15.00
https://doi.org/10.1145/3453154

number of potential limitations. There are, for example, a number of application scenarios, where the user preferences cannot be reliably estimated from their past interactions. This is often the case with high-involvement products (e.g., when recommending a smartphone), where we even might have no past observations at all. Furthermore, what to include in the set of recommendations can be highly context dependent, and it might be difficult to automatically determine the user's current situation or needs. Finally, another assumption often is that users already know their preferences when they arrive at the site. This might, however, not necessarily be true. Users might also *construct* their preferences only during the decision process [152], when they become aware of the space of the options. In some cases, they might also learn about the domain and the available options only during the interaction with the recommender [154].

The promise of **Conversational Recommender Systems (CRS)** is that they can help to address many of these challenges. The general idea of such systems, broadly speaking, is that they support a task-oriented, multi-turn dialogue with their users. During such a dialogue, the system can elicit the detailed and current preferences of the user, provide explanations for the item suggestions, or process feedback by users on the made suggestions. Given the significant potential of such systems, research on CRS already has some tradition. Already in the late 1970s, Rich [127] envisioned a computerized librarian that makes reading suggestions to users by interactively asking them questions, in *natural language*, about their personality and preferences. Besides interfaces based on **natural language processing (NLP)**, a variety of *form-based user interfaces*[1] were proposed over the years. One of the earlier interaction approaches in CRS based on such interfaces is called *critiquing*, which was proposed as a means for query reformulation in the database field already in 1982 [144]. In critiquing approaches, users are presented with a recommendation soon in the dialogue and can then apply pre-defined critiques on the recommendations, e.g., ("less $$") [15, 49].

Form-based approaches can generally be attractive as the actions available to the users are pre-defined and non-ambiguous. However, such dialogues may also appear non-natural, and users might feel constrained in the ways they can express their preferences. NLP-based approaches, however, for a long time suffered from existing limitations, e.g., in the context of processing voice commands. In recent years, however, major advances were made in language technology. As a result, we are nowadays used to issuing voice commands to our smartphones and digital home assistants, and these devices have reached an impressive level of recognition accuracy. In parallel to these developments in the area of voice assistants, we have observed a fast uptake of *chatbot* technology in recent years. Chatbots, both rather simple and more sophisticated ones, are usually able to process natural language and are nowadays widely used in various application domains, e.g., to deal with customer service requests.

These technological advances led to an increased interest in CRS during the last years. In contrast to many earlier approaches, we however observe that today's technical proposals are more often based on machine learning technology instead of following pre-defined dialogue paths. However, often there still remains a gap between the capabilities of today's voice assistants and chatbots compared to what is desirable to support truly conversational recommendation scenarios [117], in particular when the system is voice controlled [161, 165].

In this article, we review the literature on CRS in terms of common building blocks of a typical conceptual architecture of CRS. Specifically, after providing a definition and a conceptual architecture of a CRS in Section 2, we discuss (i) interaction modalities of CRS (Section 3), (ii) the knowledge and data they are based upon (Section 4), and (iii) the computational tasks that have

[1] With form-based UIs, we refer to systems approaches where users fill out forms and use check boxes or buttons.

to be accomplished in a typical CRS (Section 5). Afterwards, we discuss evaluation approaches for CRS (Section 6) and finally give an outlook on future directions.

2 DEFINITIONS AND RESEARCH METHODOLOGY

In this section, we discuss relevant preliminaries to our work. First, we provide a general characterization and conceptual model of CRS. Second, we discuss our research methodology.

2.1 Characterization of Conversational Recommender Systems

There is no widely established definition in the literature of what represents a CRS. In this work, we use the following definition.

Definition 2.1 (Conversational Recommender System–CRS). A CRS is a software system that supports its users in achieving recommendation-related goals through a multi-turn dialogue.

One fundamental characteristic of CRS is their task-orientation, i.e., they support recommendation specific tasks and goals. The main task of the system is to provide recommendations to the users, with the goal to support their users' decision-making process or to help them find relevant information. Additional tasks of CRS include the acquisition of user preferences or the provision of explanations. This specific task orientation distinguishes CRS from other dialogue-based systems, such as the early ELIZA system [158] or similar *chat robot* systems [151].

The other main feature of a CRS according to our definition is that there is a *multi-turn* conversational interaction. This stands in contrast to systems that merely support question-answering (Q&A tools). Providing one-shot Q&A-style recommendations is a common feature of personal digital assistants like Apple's Siri and similar products. While these systems already today can reliably respond to recommendation requests, e.g., for a restaurant, they often face difficulties maintaining a multi-turn conversation. A CRS therefore explicitly or implicitly implements some form of *dialogue state management* to keep track of the conversation history and the current state.

Note that our definition does not make any assumptions regarding the *modality* of the inputs and the outputs. CRS can be voice controlled, accept typed text, or obtain their inputs via form fields, buttons, or even gestures. Likewise, the output is not constrained and can be voice, speech, text, or multimedia content. No assumptions are also made regarding who drives the dialogue.

Generally, conversational recommendation shares a number of similarities with *conversational search* [115]. In terms of the underlying tasks, search and recommendation have in common that one main task is to rank the objects according to their assumed relevance, either for a given query (search) or the preferences of the user (recommendation). Furthermore, in terms of the conversational part, both types of systems have to interpret user utterances and disambiguate user intents in case natural language interactions are supported. In conversational search systems, however, the assumption often is that the interaction is based on "written or spoken form" [115], whereas in our definition of CRS various types of input modalities are possible. Overall, the boundary between (personalized) conversational search and recommendation systems often seems blurry, see References [86, 139, 172], in particular as often similar technological approaches are applied. In this survey, we limit ourselves to works that explicitly mention recommendation as one of their target problems.

2.2 Conceptual Architecture of a CRS

A variety of technical approaches for building CRS were proposed in the past two decades. The specifics of the technical architecture of such solutions depend on the system's functionality, i.e., whether or not voice input is supported. Still, a number of typical interoperating conceptual components of such architectures can be identified, as shown in Figure 1.

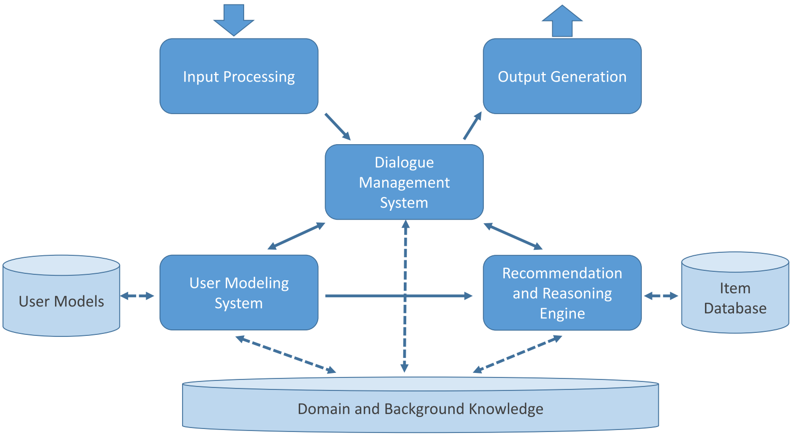
Input Processing
Output Generation
Dialogue Management System
User Models
User Modeling System
Recommendation and Reasoning Engine
Item Database
Domain and Background Knowledge

Fig. 1. Typical architecture of a conversational recommender system (see also Reference [142]).

Computational Elements. One central part of such an architecture usually is a *Dialogue Management System* (also called "state tracker" or similarly in some systems). This component drives the process flow. It receives the processed inputs, e.g., the recognized intents, entities and preferences, and correspondingly updates the dialogue state and user model. After that, using a recommendation and reasoning engine and background knowledge, it determines the next action and returns appropriate content like a recommendation list, an explanation, or a question to the output generation component.

The *User Modeling System* can be a component of its own, in particular when there are long-term user preferences to be considered, or not. In some cases, the current preference profile is implicitly part of the dialogue system. The *Recommendation and Reasoning Engine* is responsible for retrieving a set of recommendations, given the current dialogue state and preference model. This component might also implement other complex reasoning functionality, e.g., to generate explanations or to compute a query relaxation (see later). Besides these central components, typical CRS architectures comprise modules for input and output processing. These can, for example, include speech-to-text conversion and speech generation. On the input side—in particular in the case of natural language input—additional tasks are usually supported, including *intent detection* and *named entity recognition* [66, 99], for identifying the users' intentions and entities (e.g., attributes of item) in their utterances.

Knowledge Elements. Various types of knowledge are used in CRS. The *Item Database* is something that is present in almost all solutions, representing the set of recommendable items, sometimes including details about their attributes. In addition to that, different types of *Domain and Background Knowledge* are often leveraged by CRS. Many approaches explicitly encode *dialogue knowledge* in different ways, e.g., in the form of pre-defined dialogue states, supported user intents, and the possible transitions between the states. This knowledge can be general or specific to a particular domain. The knowledge can furthermore either be encoded by the system designers or automatically learned from other sources or previous interactions. A typical example for learning approaches are those that use machine learning to build statistical models from corpora of recorded dialogues. Generally, domain and background knowledge can be used by all computational elements. Input processing may need information about entities to be recognized or knowledge about the pre-defined intents. The user modeling component may be built on estimated interest weights regarding certain item features, and the reasoning engine may use explicit inference knowledge to derive the set of suitable recommendations.

2.3 Research Method: Identifying Relevant Works

We followed a semi-systematic approach to identify relevant papers. We first queried several digital libraries[2] using pre-defined search strings such as "conversational recommender system," "interactive recommendation," "advisory system," or "chatbot recommender." The returned papers were then manually checked for their relevance based on titles and abstracts. Papers considered relevant were read in detail and, if considered to be in the scope of the paper, used as a starting point for a snowballing procedure. Overall, the paper selection process surfaced 121 papers on CRS that we considered in this work.[3] Looking at the type of these papers, the majority of the works described technical proposals for one of the computational components of a CRS architecture. A smaller set of papers described demo systems. Another smaller set were analytical ones that, for example, reviewed certain general characteristics of CRS.

Generally, we only included papers that are compliant with our definition of a CRS given above. We therefore did not include papers that discussed one-shot or multi-step question-answering systems [133, 166], even when the question or task was about a recommendation. We also did not consider general dialogue systems like chatbot systems, which are not task-oriented, or systems that only support a query-response interaction process like a search engine without further dialogue steps, e.g., Reference [31]. Furthermore, we did not include dialogue-based systems, which were task-oriented, but not on a recommendation task, e.g., the end-to-end learning approaches presented in Reference [159] and Reference [76], which focus on restaurant search and movie-ticket booking. Furthermore, we excluded a few works like Reference [50] or Reference [174], which use the term "interactive recommendation," which however rather refers to a system that addresses observed user interest changes over time, but is not designed to support a dialogue with the user. Other works like Reference [138] or Reference [174] mainly focus on finding good strategies for acquiring an initial set of ratings for cold-start users. While these works can be seen as supporting an interactive process, there is only one type of interaction, which is furthermore mostly limited to a profile-building phase. Finally, there are a number of works where users of a recommendation systems are provided with mechanisms to fine-tune their recommendations, which is sometimes referred to as "user control" [61]. Such works, e.g., Reference [163], in principle support user actions that can be found in some CRS, for example to give feedback on a recommendation. The interaction style of such approaches is however not a dialogue with the system.

3 INTERACTION MODALITIES OF CRS

The recent interest in CRS is spurred both by developments in NLP and technological advances such as broadband mobile internet access and new devices like smartphones and home assistants. Our review of the literature however shows that the interaction between users and a CRS is neither limited to natural language input and output nor to specific devices.

3.1 Input and Output Modalities

The majority of the surveyed papers explicitly or implicitly support two main forms of inputs and outputs, either as the only modality or combined in a hybrid approach:

- Based on forms and structured layouts, as in a traditional web-based (desktop) application.
- Based on natural language, either in written or spoken form.

[2] We looked at Springer Link, the ACM Digital Library, IEEE Xplore, ScienceDirect, arXiv.org, and ResearchGate.

[3] A few additional papers were considered later on based on reviewer suggestions. Overall, while this research is not the result of a *systematic* literature review, we are confident that the selection of considered papers is not biased, given our structured and documented search process.

Approaches that are *exclusively* based on forms (including buttons, radio-buttons, etc.) and structured text for the output are common for almost all (except Reference [47]) critiquing-based approaches, e.g., References[7, 39, 52, 89, 123], as well as for web-based interactive advisory solutions as presented, e.g., in References [41, 55, 60]. In such applications, users typically follow pre-defined dialogue paths and interact with the application by filling forms or choosing from pre-defined options. The final output typically is a structured visual representation in the form of a list of options.

However, approaches that are *entirely* based on natural language interactions include task-oriented dialogue systems like the early proposal from Reference [127], the explanation-aware conversational system proposed in Reference [109], as well as more recent (deep) learning-based approaches, e.g., References [46, 48, 77]. *Spoken-text-only* approaches are often implemented on smart speakers like Amazon Alexa or Google Home, e.g., References [4, 36]. Compared to form-based approaches, these solutions usually offer more flexibility in the dialogue and sometimes support chit-chat and mixed-initiative dialogues. Major challenges can, however, lie in the understanding of the users' utterances and the identification of their intents. But also the presentation of the recommendations can be difficult, in particular when more than one option should be provided at once.

Hybrid approaches that combine natural language with other modalities are, therefore, not uncommon. For example, systems that support written natural language dialogues often rely on list-based or other visual approaches to present their results [73, 172]. The work presented in Reference [167], supports a hybrid visual/natural language interaction mechanism, where recommendations are displayed visually, and users can provide feedback to certain features in a critiquing-like form in natural language. Yet other systems support voice input, but present the recommendations in textual form [47, 142], because it can be difficult to present more than one recommendation at a time through spoken language without overwhelming the users. *Chatbot* applications, finally, often combine natural language input and output with structured form elements (e.g., buttons) and a visually-structured representation of the recommendations [53, 62, 100, 114].

Besides written or spoken language and fill-out forms, a few other alternative and *application-specific modalities* for inputs and outputs can be found. The dialogue system presented in Reference [150], for example, supports multiple types of inputs, including *visual inputs* on a geographic map, *pen gestures* like zooming, or *handwritten input*. The work proposed in Reference [18] furthermore tries to process *non-verbal* input, like *body postures*, *gestures*, *facial expressions*, as well as *speech prosody* to estimate the user's emotions and attitudes to acquire implicit feedback and preferences.

In terms of the *outputs*, several approaches use interactive geographic *maps*, often as part of a multi-modal output strategy [5, 39, 73, 150]. The applicability of map-based approaches is limited to certain application domains, e.g., travel and tourism, but can help to overcome various challenges regarding the user experience with conversational systems [125]. The use of **embodied conversational agents (ECAs)** [19] as an additional output mechanism is also not uncommon in the literature [41, 52] because of the assumed general persuasive potential of humanlike avatars [2, 38]. Various factors can impact the effectiveness of such ECAs. In Reference [43], for example, the authors analyze the effects of *non-verbal* behavior (e.g., the facial expressions) on the effectiveness of an ECA in the context of a dialogue-based recommender system. Research on the specific effects of using different variants of an ECA in the context of recommender systems is, however, generally rare.

Finally, a few works exist where users interact with a recommendation system within a *virtual, three-dimensional space*. In References [33, 34], the authors describe a virtual shopping environment where users interact with a critiquing-based recommender and can, in addition, collaborate with other users. Supporting group decisions is also the goal of the work presented in Reference [1]. In this work, however, no three-dimensional (3D) visualization is supported, and the focus of

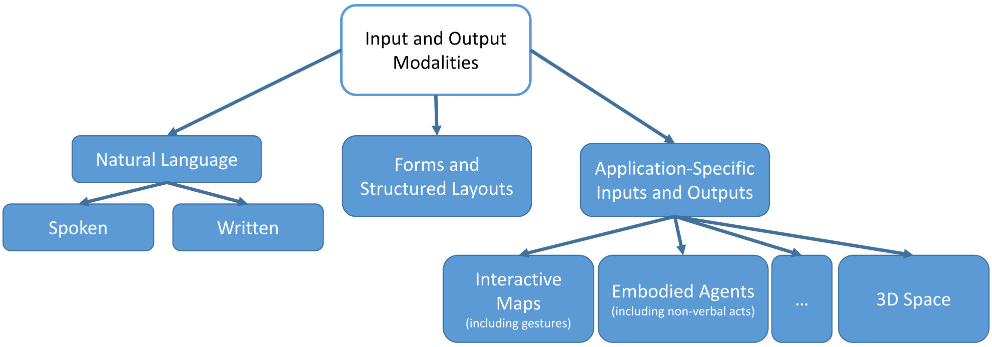
Input and Output Modalities
Natural Language
Forms and Structured Layouts
Application-Specific Inputs and Outputs
Spoken
Written
Interactive Maps (including gestures)
Embodied Agents (including non-verbal acts)
...
3D Space

Fig. 2. Categorization of input and output modalities.

the work is mostly to enable the conversation between a group of users supported by a recommender system. Figure 2 provides an overview of common input and output modalities found in the literature.

3.2 Application Environment

Stand-alone and Embedded Applications. CRS can both be stand-alone applications or part of a larger software solution. In the first case, recommendation is the central functionality of the system. Examples for such applications include the mobile tourist guides proposed in References [7, 60, 86], the interactive e-commerce advisory systems discussed in References [41, 58], or the early *FindMe* browsing and shopping systems [14, 15]. In the second case, that of an embedded application, the CRS does not (entirely) stand on its own. Often, the CRS is implemented in the form of a chatbot that is embedded within e-commerce solutions [32, 164] or other types of web-portals [21]. In some cases, the CRS is also part of a multi-modal 2D or 3D user experience, like in Reference [33] and Reference [43]. A special case in this context is the use of a CRS on voice-based home assistants (smart speakers) [4, 36]. In such settings, providing recommendations is only one of many functionalities the device is capable of. Users might therefore not actually perceive the system as primarily being a recommender.

Supported Devices. An orthogonal aspect regarding the application environment of a CRS is that of the supported devices. This is particularly important, because the specific capabilities and features of the target device can have a significant impact on the design choices when building a CRS. The mentioned smart speaker applications, for example, are specifically designed for hardware devices that often only support voice-based interactions. This can lead to specific challenges, e.g., when it comes to determining the user's intent or when a larger set of alternatives should be presented to the users. The interaction with chatbot applications, however, is typically not tied to specific hardware devices. Commonly, they are either designed as web applications or as smartphone and tablet applications. However, the choice of the used communication modality can still depend on the device characteristics. Typing on small smartphone screens may be tedious and the limited screen space in general requires the development of tailored user interfaces.

The applicability of CRS is not limited to the mentioned devices. Alternative approaches were, for example, investigated in References [18, 37]. Here, the idea is that the CRS is implemented as an application on an interactive wall that could be installed in a real store. A camera is furthermore used to monitor and interpret the user's non-verbal communication actions, in particular facial expressions and gestures. An alternative on-site environment was envisioned in Reference [170]. Here, the ultimate goal is to build a CRS running on a service robot, in this case one that is able to elicit a customer's food preferences in a restaurant. Yet another application scenario, that of future in-car recommender systems, is sketched in Reference [83]. Given the specific situation in a driving scenario, the use of speech technology often is advisable [22], which almost naturally leads to conversational recommendation approaches, e.g., for driving-related aspects like navigation or entertainment [8, 9].

3.3 Interaction Initiative

A central design question for most conversational systems is who takes the initiative in the dialogue. Traditionally, we can differentiate between (i) system-driven, (ii) user-driven, and (iii) mixed-initiative systems. When considering CRS primarily as dialogue systems, such a classification can in principle be applied as well, but the categorization is not always entirely clear.

Critiquing-based systems are often considered to be mainly *system-driven*, and sometimes mixed-initiative, e.g., in Reference [148]. In such applications, the users are typically first asked about their preferences, e.g., using a form, and then an initial recommendation is presented. Users can then use a set of pre-defined or dynamically determined critiques to further refine their preferences. While the users in such applications have some choices regarding the dialogue flow, e.g., they can decide to accept a recommendation or further apply critiques, these choices are typically very limited and the available critiques are determined by the system. Another class of mostly system-driven applications are the form-based interactive advisory systems discussed in Reference [41]. Here, the system guides the user through a personalized preference elicitation dialogue until enough is known about the user. Only after the initial recommendations are displayed, the user can influence the dialogue by selecting from pre-defined options like asking for an explanation or by relaxing some constraints.

The other extreme would be a *user-driven* system, where the system takes no proactive role. The resulting dialogue therefore consists of "user-asks, system-responds" pairs, and it stands to question if we would call such an exchange a conversational recommendation. Such conversation patterns are rather typical for one-shot query-answering, search and recommendation systems that are not in the scope of our survey. As a result, in the papers considered relevant for this study, we did not find any paper that aimed at building an *entirely* user-driven system in which the system never actively engages in a dialogue, e.g., when it does not ask any questions ever. A special case in that context is the recommender system proposed in Reference [82], which monitors an ongoing group chat and occasionally makes recommendations to the group based on the observed communication.

This observation is not surprising because every CRS is a task-oriented system aiming to achieve goals like obtaining enough reliable information about the user's preferences. As a result, almost all approaches in the literature are *mixed-initiative* systems, although with different degrees of system guidance. Typical chatbot applications, for example, often guide users through a series of questions with pre-defined answer options (using forms and buttons), and at the same time allow them to type in statements in natural language. In fully NLP-based interfaces, users typically have even more freedom to influence how the dialogue continues. Still, also in these cases, the system typically has some agenda to move the conversation forward.

Technically, even a fully NLP-based dialogue can almost entirely be system-driven and mostly rely on a "system asks, user responds" [172] conversation pattern. Nonetheless, the provision of a natural language user interface might leave the users disappointed when they find out that they can never actively engage in the conversation, e.g., by asking a clarification question or explanation regarding the system's question.

3.4 Discussion

A variety of ways exist in which the user's interaction with a CRS can be designed, e.g., in terms of the input and output modalities, the supported devices, or the level of user control. In most surveyed papers, these design choices are, however, rarely discussed. One reason is that in many cases the proposed technical approach is mostly independent of the interaction modality, e.g., when the work is on a new strategy to determine the next question to ask to the user. In other cases, the modalities are pre-determined by the given research question, e.g., how to build a CRS on a mobile.

More research therefore seems required to understand how to make good design choices in these respects and what the implications and limitations of each design choice are. Regarding the chosen form of inputs and outputs, it is, for example, not always entirely clear if natural language interaction makes the recommendation more efficient or effective compared to form-based inputs. Pure natural language interfaces in principle provide the opportunity to elicit preferences in a more natural way. However, these interfaces have their limitations as well. The accuracy of the speech recognizer, for example, can have a major impact on the system's usability. In addition, some users might also be better acquainted and feel more comfortable with more traditional interaction mechanisms (forms and buttons). According to the study in Reference [54], a mix of a natural language interface and buttons led to the best user experience. Moreover, in Reference [102], it turned out that in situations of disambiguation, i.e., when a user has to choose among a set of multiple alternatives, mixed-interaction mode (NLP interface with buttons) can make the task easier for users. Overall, while in some cases the choice of the modalities is predetermined through the device, finding an optimal combination of interaction modalities remains challenging, in particular as individual user preferences might play a role here.

More studies are also needed to understand how much flexibility in the dialogue is required by users or how much active guidance by the system is appreciated in a certain application. Furthermore, even though language-based and in particular voice-based conversations have become more popular in recent years, certain limitations remain. It is, for example, not always clear how one would describe a set of recommendations when using voice output. Reading out more than one recommendation seems impractical in most cases and something that we could call "recommendation summarization" might be needed.

Despite these potential current limitations, we expect a number of new opportunities where CRS can be applied in the future. With the ongoing technological developments, more and more devices and machines are equipped with CPUs and are connected to the internet. In-store interactive walls, service robots and in-car recommenders, as discussed above, are examples of visions that are already pursued today. These new applications will, however, also come with their own general challenges (e.g., privacy considerations, aspects of technology acceptance) and application-specific ones (e.g., safety considerations in an in-car setting).

4 UNDERLYING KNOWLEDGE AND DATA

Depending on the chosen technical approach, CRS have to incorporate various types of knowledge and background data to function. Clearly, like any recommender, there has to be information about the recommendable items. Likewise, the generation of the recommendations is either based on explicit knowledge, for example recommendation rules or constraints, or on machine learning models that are trained on some background data. However, conversational systems usually rely on additional types of knowledge about the user intents that the CRS supports, the possible states in the dialogue, or data such as recorded and transcribed natural language recommendation dialogues that are used to train a machine learning model. In the following sections, we provide an

Table 1. High-level Overview of Selected Domain-independent User Intents Found in the Literature

Intent Name	Intent Description
Initiate Conversation	Start a dialogue with the system.
Chit-chat	Utterances unrelated to the recommendation goal.
Provide Preferences	Share preferences with the system.
Revise Preferences	Revise previously stated preferences.
Ask for Recommendation	Obtain system suggestions.
Obtain Explanation	Learn more about why something was recommended.
Obtain Details	Ask about more details of a recommended object.
Feedback on Recommendation	Give feedback on the provided recommendation(s).
Restart	Restart the dialogue.
Accept Recommendation	Accept one of the recommendations.
Quit	Terminate the conversation.

overview on the different types of knowledge and data that were used in the literature to build a CRS.

4.1 User Intents

CRS are dialogue systems designed to serve very specific purposes in the context of information filtering and decision making. Therefore, they have to support their users' particular information needs and intents that can occur in such conversations. In many CRS, the set of user intents that the system supports is pre-defined and represents a major part of the manually engineered background knowledge on which the system is built. In particular in NLP-based approaches, detecting the current user's intent and selecting the system's response is one of the main computational tasks of the system, see also Section 5. In this section, we will therefore mainly focus on NLP-based systems.

The set of user intents that the system supports varies across the different CRS that are found in the literature, and the choice of which intents to support ultimately depends on the requirements of the application domain. However, while a subset of the intents that the system supports is sometimes specific to the application as well, there are a number of intents that are common in many CRS. In Table 1, we provide a high-level overview of *domain-independent* user intents that we have found in our literature review. The order of the intents in Table 1 roughly follows the flow of a typical recommendation dialogue. This overview is also intended to serve as a tool for the designers of CRS to check if there are any gaps in their current system with respect to potential user needs that are not well supported.

Research on what are relevant user intents is generally scarce, and we only found 11 papers that explicitly discussed user intents. Among these 11, only few of them, e.g., see References [16, 100, 164], considered the majority of the domain-independent intents shown in Table 1. Others, like References [65, 105, 142], only discuss certain subsets of them. Yet another set of papers focused on very application-specific intents in the context of group recommendation [1, 103].

Starting, re-starting, and ending the dialogue. In NLP-based CRS, either the system or the user can initiate the dialogue. In a user-driven conversation, the recommendation seeker might, for example, explicitly ask for help [100] or make a recommendation request [162] to start the interaction. One typical difficulty in this context is to recognize such requests when the dialogue starts with chit-chat. Once the recommendation dialogue is moving on, it is not uncommon that users want to start over, i.e., begin the session from scratch and "reset their profile" [100]. Previous studies found that such an intent was found in 5.2% of the dialogues [16] or that 36.4% of the users had this intent in a conversation [65]. Finally, at the end of the conversation, the user has either found a recommendation useful and accepts it in some form (e.g., by purchasing or consuming an item) or not. In either case, the CRS has to react to the intent in some form by redirecting the user accordingly, e.g., to the shopping basket, or by saying goodbye.

Chit-chat. Many NLP-based systems support chit-chat in the conversation. In the study in Reference [164], nearly 80% of the recorded user utterances were considered chit-chat. This number indicates that supporting chit-chat conversations can be a valuable means to create an engaging user experience. Furthermore, the study in Reference [164] showed that chit-chat can also help to reduce user dissatisfaction, even though this part of the conversation is irrelevant to achieving the interaction goal.

Preference Elicitation. Understanding the user's preferences is a key task for any CRS. Preference information can be provided by the user in different ways. In an initial phase of the dialogue, the user might specify some of the desired characteristics of the item that she or he is interested in or even provide strict filtering constraints. In Reference [105], this process is termed as "give criteria." In later phases, the user might however also want to revise the previously stated preferences. Note that some authors also consider *answering*—to a system-provided question or proposal for a constraint [25, 145]—as a dialogue intent during preference elicitation [156]. Since in NLP-based systems a user may respond in an arbitrary way, it is clearly important for the system to disambiguate an answer by the user from other utterances. Such an "Answer" intent nonetheless is different from the other intents discussed here, as the intent is a response to the system's initiative of asking.

Also later in the process, preferences can be stated by the user in different ways after an initial recommendation is made by the system. In critiquing-based approaches, the users can, for example, add additional constraints in case the choice set is too large, relax some of the previously stated ones, or state that they already know the item [56, 123, 142]. Generally, a system might also allow the user to inspect, modify, and delete the current profile (supporting a "show profile" intent) [100]. By analyzing the interaction logs of a prototypical voice-controlled movie recommender, e.g., in Reference [65], the authors found that many users (41.1%) at some stage try to refine their initially stated preferences. In particular in case of unsatisfactory system responses, some users might furthermore also have the intent to "reject" [156] a recommendation or "restate" their preferences. In the study presented in Reference [16], this, however, happened only in 1.5% of the interactions.

Obtaining Recommendations and Explanations. There are various ways in which users might ask for recommendations and additional information about the items. Asking for recommendations often happens at the very beginning of a dialogue, but this event can also occur after the user has revised the preferences. In case a currently displayed list of options is not satisfactory, users also might ask the system to "show more" options [16] or ask for a similar item for comparison. For each of the items, the user might want to learn more about its details or ask for an explanation, e.g., why it was recommended [100]. Finally, an alternative form of requesting a recommendation is to ask the system about its opinion ("how about") regarding a certain item, see e.g., Reference [164].

4.2 User Modeling

The interactive elicitation of the user's *current* preferences or needs and constraints is another central task of CRS. As discussed above, this can be done through different modalities and by supporting various ways for users to express what they want. The acquired preferences are typically recorded in explicit form within a *user profile*, based on which further inferences regarding the relevance of individual items can be made. There are two main ways of representing the preference information in such explicit models:

- Preference expressions or estimates regarding *individual items*, e.g., ratings, like and dislike statements, or implicit feedback signals. In Reference [39], for example, users are initially presented with a number of tourism destinations and asked which of them match their preferences.
- Preferences regarding *individual item facets*. These facets can either relate to item attributes (e.g., the genre of a movie) or the desired functionalities.

For the latter class of approaches, the goal of the CRS is sometimes referred to as "slot filling," i.e., the recommender seeks to obtain values for a set of pre-defined facets. Sometimes, also the preference *strength*, e.g., must or wish, can be relevant [123]. While different approaches for mining possible facets from structured and unstructured text documents were proposed in the literature [157], the set of facets is often manually engineered based on domain expertise. Furthermore, in case the facets refer to functional requirements as in References [55, 160], additional knowledge has to be encoded in the system to match these requirements with the recommendable items. In Reference [55], for example, the user of an interactive camera recommender is asked about the type of photos she or he wants to take (e.g., sports photography), and a constraint-based system is then used to determine cameras that are suited for this purpose.

Finally, a few works exist that do *not* assume the existence of a set of engineered item features. In Reference [119], for example, an approach is proposed for preference elicitation, where user repeatedly specify preferences on items and the system then finds items that are similar in terms of unstructured features like keywords or tags. Similarly, other types of non-engineered features (tags, key phrases, or latent representations) were used in the preference elicitation approaches proposed in References [81, 84] and Reference [149].

Besides such ephemeral user models that are constructed during the ongoing session, some approaches in the literature also maintain long-term preference profiles [4, 123, 142, 154]. In the critiquing approach in Reference [123], for example, the system tries to derive long-term and supposedly more stable preferences (e.g., for non-smoking rooms in restaurants) from multiple sessions. In the content-based recommendation approach adopted in Reference [142], a probabilistic model is maintained based on past user preferences for items. In general, a key problem when recommending based on two types of models (long-term and short-term) is to determine the relative importance of the individual models. One so far unexplored option could lie in the consideration of contextual factors such as seasonal aspects, the user's location, or the time of the day.

Finally, there are also approaches that try to leverage information about the collective preferences of a user community, in particular for cold-start situations [101]. If nothing or little is known yet about the user's preferences, then a common strategy is to recommend popular items, where item popularity can be determined based on user ratings, reviews, or past sales numbers as in Reference [47]. The feedback obtained for these popular items can then be used to further refine the user model.

4.3 Dialogue States

To be able to support a multi-turn, goal-oriented dialogue with their users, CRS have to implement appropriate means to keep track of the state of the dialogue to decide on the next *conversational move*, i.e., the next action. In many CRS implementations and in particular in knowledge-based approaches, dialogue management is based on a finite state machine, which not only defines the possible states but also the allowed transitions between the states [85, 86, 153, 155]. In the Advisor

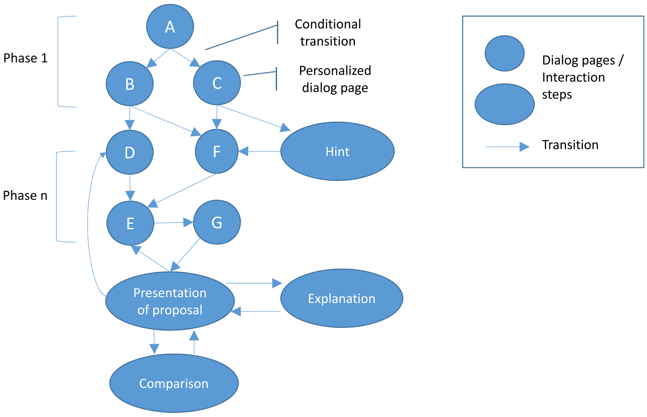

Fig. 3. Pre-defined dialogue states in the Advisor Suite system (adapted from Reference [58]).

Suite framework [55, 58], for example, the entire recommendation logic including the dialogue flow was modelled with the help of graphical editors.

Figure 3 shows a schematic overview of such a dialogue model. It consists of (i) a number of dialogue steps to acquire the user's preferences through questions and (ii) special dialogue states in which the system presents the results, provides explanations, or shows a comparison between different alternatives. The possible transitions are defined at design time, but which path is taken during the dialogue is determined dynamically based on decision rules. Another example for a work that is based on a pre-defined set of states and possible transitions is the interactive tourism recommender system proposed in Reference [86]. In their case, the transitions at runtime are not determined based on manually engineered decision rules, but learned from the data using reinforcement learning techniques, where one goal is to minimize the number of required interaction steps.

Technically, there are different ways of explicitly representing such state machines. Some tools, as the one mentioned above, use visual representations, others rely on textual and declarative representations like "dialogue grammars" [13] and case-frames [12]. Google's DialogFlow,[4] as an example of a commercial service, uses a visual tool to model linear and non-linear conversation flows, where non-linear means that there are different execution paths, depending on the user's responses or contextual factors. Finally, in some cases, the possible states are simply hard-coded as part of the general program logic of the application.

In some works, and in particular in early critiquing-based ones that are based on forms and buttons [122, 123, 134], only a few generic dialogue states exist, which means that no complex flow has to be designed. After an initial preference elicitation stage, recommendations are presented, and the system offers a number of critiques that the user can apply until a recommendation is accepted or rejected. Dialogue state management is therefore in some ways relatively light-weight. The main task of the system in terms of dialogue management is to keep track of the user responses and, in case of dynamic critiquing, make inferences regarding the next critiques to offer.

Similarly, in some NLP-based conversational preference elicitation systems such as References [29, 172], there are mainly two phases: asking questions, in this case in an adaptive way, and presenting a recommendation list. In other NLP-based systems, the possible dialogue states are not modeled explicitly as such, but implicitly result from the implemented intents. For example,

[4] https://dialogflow.com/.

whether or not there is a dialogue state "provide explanation" depends on the question whether a corresponding intent was considered in the design phase of the system.

Finally, in the NLP-based end-to-end learning CRS proposed in Reference [75], the dialogue states are in some ways also modeled implicitly, but in a different way. This system is based on a corpus of recorded human conversations (between crowdworkers) centered around movie recommendations. This corpus is used to train a complex neural model, which is then used to react to utterances by users. Looking at the conversation examples, these conversations, besides some chit-chat, mainly consist of interactions where one communication partner asks the other if she or he likes a certain movie. The sentiment of the answer of the movie seeker is then analyzed to make another recommendation, again mostly in the form of a question. The dialogue model is therefore relatively simple and encoded in the neural model. It seemingly does not support many other types of intents or information requests that do not contain movie names (e.g., *"I would like to see a sci-fi movie"*).

4.4 Background Knowledge

Besides the discussed knowledge regarding the set of supported user intents or the possible dialogue states, CRS are based on additional types of knowledge and data. This knowledge for example includes information related to the items (e.g., attributes and ratings), corpora of logged natural language conversations for learning, and additional knowledge sources used for entity recognition.

4.4.1 Item-related Information. Like any recommender, also a conversational system has to have access to a database with information about the recommendable items. Such a database can contain item ratings, metadata that can be presented to the user (e.g., the genre of a movie or the director), community-provided tags, or extracted keyphrases. These item attributes can furthermore serve as a basis for other computational tasks, e.g., to compute the personalized recommendations, to generate explanations, or to determine which questions can be asked to the user.

In the examined papers, we found that researchers used a number of different databases. Some works are based on typical rating datasets, e.g., from MovieLens or Netflix, whereas other researchers created their own datasets or relied on preexisting datasets from different domains. In Table 2, we provide examples of datasets containing item-related information. It can be observed that it is not uncommon, e.g., in critiquing-based applications, that researchers solely rely on datasets that they created or collected for the purpose of their studies, i.e., there is limited reuse of datasets by other researchers. One main underlying reason is that in most papers we analyzed, researchers did not publicly share their datasets.

4.4.2 Dialogue Corpora Created to Build CRS. NLP-based dialogue systems are usually based on training data that consist of recorded and often annotated conversations between humans (interaction histories). A number of initiatives were therefore devoted to create such datasets that can be used to build CRS. Other researchers, in contrast, rely on dialogue datasets that were created or collected for other purposes. Generally, these corpora can be obtained with the help of crowdworkers [64, 75, 139], by annotating interviews [16, 18, 109], or by logging interactions with a chatbot like in Reference [63]. Table 3 shows examples of such datasets used in recent research.

Note that in some cases when building a CRS, these dialogue corpora are combined with other knowledge bases [75, 170]. In Reference [75], for example, both a dialogue corpus and MovieLens data are used for the purposes of sentiment analysis and rating prediction. Such a combination of datasets can be necessary when there is not enough relevant information in the dialogues.

Table 2. Examples of Datasets Containing Item-related Information

Domain	Description
Movies	Traditional movie rating databases from MovieLens, EachMovie, Netflix, used for example in References [75, 174, 174].
Electronics	A product database with more than 600 distinct products was collected from various retailers [47].
	A smartphone database consisting of 1721 products with multiple features [34].
	An Amazon electronics review dataset containing millions of products, user reviews and product metadata [172].
	A dataset consisting of 120 personal computers, each with 8 features [134].
Travel	More than 100 sightseeing spots in Japan with 25 different features [53].
	A database of restaurants in the San Francisco area covering 1,900 items with multiple features like cuisine, ratings, price, location, or parking [142].
	Search logs and reviews of 3,549 users of a restaurant review provider, focusing on locations in Cambridge [29].
	A travel destinations dataset, crawled from online platforms containing 5,723,169 venues in 180 cities around the globe [39].
	A restaurants dataset crawled for Dublin city, which consists of 632 restaurants with 28 different features [92].
Food Recipes	A food recipe dataset containing dishes and their ingredients [170].
E-commerce	A product database of 11M products and logged data from the search engine of an e-commerce website was collected. The logged data consists of 3,146,063 unique questions [164].
Music	A music dataset crawled from multiple online sources, containing 2,778 songs with 206k explanatory statements and 22 user tags [173].

4.4.3 Logged Interaction Histories. Building an effective CRS requires to understand the conversational needs of the users, e.g., how they prefer to provide their preferences, which intents they might have, and so on. One way to better understand these needs is to log and analyze interactions between users and a prototypical system. These logs then serve as a basis for further research. Differently from the dialogue corpora discussed above, these datasets were often not primarily created to build a CRS, but to better understand the interaction behavior of users. In References [154, 155], for example, the interactions of the user with a specific NLP-based CRS were analyzed regarding dialogue quality and dialogue strategies. In References [16, 18], user studies were conducted prior to developing the recommender system to understand and classify possible feedback types by users. In some approaches like References [18, 114] researchers annotated and labeled such datasets for the purpose of model training and system initialization. However, such logged histories are—except for Reference [114]—typically much smaller in size than the dialogue corpora discussed above, mostly because they were collected during studies with a limited number of participants. Examples of datasets obtained by logging system interactions and user studies are shown in Table 4.

4.4.4 Lexicons and World Knowledge. Researchers often use additional knowledge bases to support the entity recognition process in NLP-based systems. In References [73, 80], for instance, information was harvested from online sources such as Wikipedia or Wikitravel to develop dictionaries for the purpose of entity-keyword mapping. Similarly, the WordNet corpus was used in Reference [73] to determine the semantic distance of an identified keyword in a conversation with predefined entities. More examples for the use of lexicons and world knowledge are shown in Table 5.

Table 3. Examples of Dialogue Corpora Created or Used to Build CRS

Domain	**Name**	**Description**
Movies	ReDial	Crowdworkers from **Amazon Mechanical Turk (AMT)** were used to collect over 10,000 dialogues centered around the theme of providing movie recommendations [75]. A paired mechanism was used where one person acts as a *recommendation seeker* and the other as a *recommender*.
	CCPE-M	A Wizard-of-Oz approach is taken to elicit movies preferences from crowdworkers within natural conversations. The dataset consists of over 500 dialogues that contain over 10,000 preference statements [116].
	GoRecDial	This dataset consists of 9,125 dialogue interactions and 81,260 conversation turns collected through pairs of human workers; here also one plays the role of a movie seeker and the other as a recommender [64].
	bAbI	In Reference [100], the authors used a general movie dialogue dataset provided by Facebook Research [40] to build a CRS. The dataset contains task-based conversations in a question-answering style. It consists of 6,733 and 6,667 dialogue conversations for training and testing respectively.
Restaurants and Travel	CRM	An initial dataset containing 385 dialogues is collected using a pre-defined dialogue template through AMT [139]. Using this dataset, a larger synthetic dataset of 875,721 simulated dialogues is created.
	ParlAI	A goal-oriented, extended version of the bAbI dataset that was collected using a bot and users. It consists of three datasets (training, development and testing), each comprising 6,000 dialogues [63].
	MultiWOZ	A large human-human dialogue corpus, which covers 7 domains and consists of 8,438 multi-turn dialogues around the themes of travel & planning recommendation [162].
Fashion	MMD	A dataset consisting of 150,000 conversations between shoppers and a large number of expert sales agents is collected. Nine dialogue states were identified in the resulting dataset [129].
Multi-domain	OpenDialKG	Chat conversation between humans, consisting of 15,000 dialogues and 91,000 conversation turns on movies, books, sports, and music [97].

Table 4. Examples of Datasets Obtained from Logged System Interactions and User Studies

Domain	**Description**
Movies	A dialogue dataset involving 347 users was collected in Reference [65] during the experimental evaluation of a recommender system.
	A subset of the ReDial dataset was analyzed and annotated in Reference [16] to classify the user feedback types in 200 dialogues at the utterance level.
	A dialogue corpus was collected in Reference [154] for the purpose of dialogue quality analysis consisting of 226 complete dialogue turns with 20 users.
	A user study was conducted in Reference [155], where a *movie seeker* and a *human recommender* converse with each other. The dialogue corpus consists of 2,684 utterances and 24 complete dialogues.
Travel	A dataset containing preferences for hotel, flight, car rental searches was collected in Reference [4] involving 200 users of a content-based recommender system that supports multiple tasks (i.e., hotel, car, flight booking) in the same dialogue.
Fashion	A user study was conducted using a virtual shopping system. A non-verbal feedback (e.g., gestures, facial expressions, voices) dataset involving 345 subjects was collected and then annotated for model training [18].
E-commerce	A dataset containing conversation logs of users with a chatbot of an online customer service center (Alibaba.com) was collected in Reference [114]. It consists of over 91,000 Q&A pairs as a knowledge base used for the information retrieval task.

Table 5. Examples of the Use of Lexicons and World Knowledge

Source Name	Description
Wikipedia	A dataset crawled from online sources (Wikipedia and Wikitravel) for the purpose of entity recognition in the travel domain [73].
WordNet	WordNet is used to compute the semantic distance between entities and keywords mentioned in the conversation [73, 80].
Wikiquote	A quote dataset crawled from two online sources, wikiquote.com and the Oxford Concise Dictionary of Proverbs [70].
Citysearch	In Reference [80], a dataset of 137,000 users reviews on 24,000 restaurants was harvested from two online sources (citysearch.com and menupages.com to generate a dictionary of mappings between semantic representations of cuisines and dialogue concepts.

4.5 Discussion

Our discussions show that CRS can be knowledge-intensive or data-intensive systems. Differently from the traditional recommendation problem formulation, where the goal is to make relevance predictions for unseen items, CRS often require much more background information than just a user-item rating matrix, in particular in the context of dialogue management.

Pre-defined Knowledge vs. Learning Approaches. In CRS approaches that use forms and buttons as the only interaction mechanism, the interaction flow is typically pre-defined in the form of the possible dialogue states, the set of supported user intents, and the user profile attributes to acquire. NLP-based systems, in contrast, are usually more dynamic in terms of the dialogue flow, and they rely on additional knowledge sources like dialogue corpora and answer templates as well as lexicon and word knowledge bases. Nonetheless, these systems typically require the manual definition of additional background knowledge, e.g., with respect to the supported user intents.

Pure "end-to-end" learning only from recorded dialogues seems challenging. In most existing approaches the set of supported interaction patterns is implicitly or explicitly predefined, e.g., in the form of "user provides preferences, systems recommends." To a certain extent, also the collection of human-to-human dialogues can be designed to support possible system responses like in Reference [75], where the crowdworkers were given specific instructions regarding the expected dialogues. As a result, the range of supported dialogue utterances can be relatively narrow. The system presented in Reference [75], for example, cannot handle a query like *"good sci-fi movie please."*

Intent Engineering and Dialogue States. In case a richer dialogue and additional functionalities are desirable, the definition of the supported user intents usually is a central and often manual task during CRS development. Compared to general-purpose dialogue systems and home assistants, however, the set of user intents that are supported is often relatively small. We have identified some common intent categories in Section 4.1. Depending on the domain, also very specific intents can be supported, e.g., asking for a style tip in a fashion recommender system [105]. Furthermore, yet another set of possible user intents has to be supported in CRS that are designed for group decision scenarios. Typical user intents can, for example, relate to the invitation of collaborators [103] or to a request for a group recommendation. Furthermore, there might be user utterances that relate to the resolution of preference conflicts and voting among group members [1, 91, 103].

Generally, the set of user intents that the system supports determines how rich and varied the resulting conversations can be. Not being able to appropriately react to user utterances can be highly detrimental to the quality perception of the system. For example, being able to explain the recommendations that the system makes is often considered as a key feature to make decision-making easier or to increase user trust in a recommender system. A user of an NLP-based system

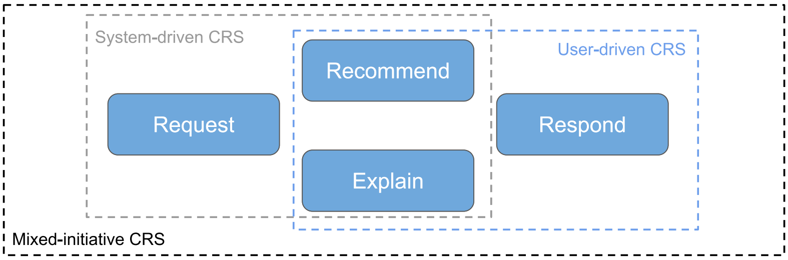

Fig. 4. Major actions taken by conversational recommender systems.

might therefore be easily disappointed by the conversation if the system fails to recognize and respond to a request for an explanation.

A key challenge, therefore, is to anticipate or learn over time which intents the users might have. Depending on the application and used technology, the design and implementation of an intent database (e.g., using Google's DialogFlow engine) can lead to substantial manual efforts and require the involvements of professional writers to achieve a certain naturalness and richness of the conversation. At the same time, the rule-based modeling approach ("if-this-then-that") as implemented by major solution providers can easily lead to large knowledge bases that are difficult to maintain, leading to a need for alternative modeling approaches [140].

5 COMPUTATIONAL TASKS

Having discussed possible user intents in recommendation dialogues, we will now review common computational tasks and technical approaches for CRS. We distinguish between (i) main tasks, i.e., those related more directly to the recommendation process, e.g., compute recommendations or determine the next question to ask, and (ii) additional, supporting tasks.

5.1 Main Tasks

Broadly speaking, CRS carry out four general types of tasks (or: system actions) during conversations [26, 101]: *Request*, *Recommend*, *Explain*, and *Respond*, see Figure 4. However, not every CRS necessarily implements all of them. System-driven CRS (as described in Section 3.3) usually drive the conversation by requesting user preferences on attributes and allowing users to give feedback on recommendations through multiple interaction cycles. User-driven systems, in contrast, can take a more passive role, and mainly respond to conversational acts by the user. In mixed-initiative systems, e.g., those based on natural language interfaces, all types of actions can be found.

5.1.1 Request. A number of CRS follow a "slot-filling" conversation approach where the system seeks to acquire preference information about a pre-defined set of item attributes or facets. One main computational task in this context is to *determine the next question to ask*, often with the goal to increase dialogue efficiency, i.e., to minimize the number of required interactions (see also Section 6). Various methods to determine the order of the facets were proposed in the literature [20, 132, 142, 170]. In an early system [142], specific weights were used to rank the item attributes for which the user has not expressed preferences yet. Entropy-based methods also consider the potential effects on the remaining item space of each attribute. They aim to identify the next question (attribute) that helps mostly to narrow down the candidate (item) space [20, 96, 104, 132, 166], sometimes including feature popularity information [96]. Considerations like this are typically also the foundation of typical *dynamic* and *compound* critiquing systems [25, 90, 111, 121, 134, 148, 171]. In compound critiquing systems, in particular, the user is not asked about feedback for one single attribute, but for more than one within one interaction, e.g., *"Different Manufacturer, Lower Processor Speed and Cheaper."* Finally, in some systems, possible sequences of questions asked to the users are pre-defined in the form of state machines [55, 58]. At runtime, the dialogue path is then chosen based on the users' inputs in the ongoing session.

Instead of using heuristics for attribute selection and static dialogue state transition rules, a number of more recent systems rely on learning-based approaches, e.g., using reinforcement learning [86, 139, 146]. In Reference [139], for example, the authors use a deep policy network to decide on the system action. Based on the current dialogue state, as modeled by a belief tracker, the system either makes a request for a pre-defined facet or generates a recommendation to be shown to the user. An alternative learning-based way to determine the question order was proposed in Reference [30]. In their work, the authors design a recommender for YouTube that leverages past watching histories of the user community and a Recurrent Neural Network architecture to rank the questions (topics) that are shown to the user in a conversational step.

An alternative to asking users about attribute-based preferences is to ask them to give feedback on selected items. This can be done either by asking them to rate individual items (e.g., by like/dislike statements) or by asking them to express their preference for item pairs or entire sets of items [81]. The computational task in this context is to determine the most informative item(s) to present to the user. Possible strategies include the selection of popular or diverse items in the cold-start phase, items that are different in terms of their past ratings or attributes, or itemsets that represent a balance of popularity and diversity [17, 93, 101, 120]. However, not only item features might be relevant for the selection of the items. In Reference [17], the authors found that a user's willingness to give feedback on an item can depend on additional factors. Specifically, they identified several situations in which the feedback probability may be higher, e.g., when the system's predicted rating deviates from the user's past experience of the item. In more recent works, again learning-based approaches are more common. The authors of References [29, 174], for example, employed bandit-based approaches to either (i) determine the next item to be shown for eliciting the user's absolute feedback (i.e., like or dislike) or (ii) to select a pair of items for obtaining the user's relative preference regarding these two items.

5.1.2 *Recommend.*

The recommendation of items is the core task of any CRS. From a technical perspective, we can find collaborative, content-based, knowledge-based, and hybrid approaches in the literature. Differently from non-conversational systems, the majority of the analyzed CRS approaches mainly relies solely on short-term preference information. However, there are also approaches that additionally consider long-term preferences of a user, e.g., to speed up the elicitation process [82, 103, 125, 130, 139, 142, 154].

In the context of critiquing-based and knowledge-based systems, different strategies are applied to filter and rank the items. For the filtering task, often constraint-based techniques [42] are applied that remove items from the candidate set that do not (exactly) match the current user's preferences. The items that remain can then be sorted in different ways [169]. In the system proposed in Reference [171], for example, the user preference model is updated after a user critique by adjusting the weights of the attributes that are involved in the critique. Then, Multi-Attribute Utility Theory [67] was used to calculate the utility of each candidate item for generating top-K recommendations for the user. An alternative ranking approach was applied in Reference [130], where a history-guided critiquing system was proposed that aims to retrieve recommendation candidates from other users' critiquing sessions that are similar to the one of the current user. In Reference [39], a critiquing-based travel recommender system was implemented that computes recommendations based on the relevance of item attributes to user preferences based on the Euclidean Distance.

Some works consider both long-term and short-term preferences of users when making recommendations [4, 82, 123, 130]. The Adaptive Place Advisor system [142] represents an early example of combining short-term and long-term preferences. Here, the user's current query is expanded by considering the probability distribution of the user's past preference for item attributes, based on her/his short-term constraints (within a conversation) and long-term constraints (over many conversations). This expanded query was then used to retrieve and rank the items for recommendation. In Reference [130], the authors proposed to leverage the successful recommendation sessions in the previous conversations to improve the efficiency of the current session (i.e., to shorten its length).

More recent works rely on machine learning models and background datasets for the recommendation task. One common approach is to train a model on the traditional user-item interaction matrix, e.g., based on probabilistic matrix factorization [29], and to then combine the user's current interactions with the trained user and item embeddings. In another approach [4], the authors rely on a content-based method based on item features and the user profile in the cold-start stage, and then switch to a Restricted Boltzmann Machine collaborative filtering method once a sufficient number of preference signals is available. In Reference [172], a hybrid multi-memory network with attention mechanism was trained to find suitable recommendations based on item embeddings and the user's query embedding. Here, the item embedding was based on the item's textual description, and the user's query embedding encoded the user's initial request and the follow-up conversations during the interaction. A hybrid model was also proposed in Reference [139], which used Factorization Machines to combine the dialogue state—represented with an LSTM-based belief tracker for each item facet—user information, and item information to train the recommendation model. In the video recommender system presented in Reference [30], finally, an RNN-based model was built for making recommendations, based on the topics selected by the users and their watching history.

In some cases, application-specific techniques were applied for the recommendation task. In References [167, 168], for example, the CRS features a visual dialogue component, where users can give feedback based on the images, e.g., *"I prefer blue color."* To implement this functionality, the system proposed in Reference [167] implemented a component that encoded item images and user feedback using a convolutional neural network, and then combined these encodings as an input to both a response encoder and a state tracker. Furthermore, various types of user behaviors (i.e., viewing, commenting, clicking) on the visually represented recommendation were considered in a bandit approach to balance exploration and exploitation.

5.1.3 Explain.

The value of explanations in general recommender systems is widely recognized [51, 106, 143]. Explanations can increase the system's perceived transparency, user trust and satisfaction, and they can help users make faster and better decisions [45]. However, according to our survey, few papers so far have studied the explanation issue specific to CRS.

In the context of critiquing-based systems, Reference [110] examined the trust-building nature of explanations. In this work, an "organization-based" explanation approach was evaluated, where the system showed multiple recommendation lists to the user, each of them labelled based on the critiquing-based selection criteria, e.g., "*cheaper but heavier.*" A more recent interactive explanation approach for a mobile critiquing-based recommender was proposed in Reference [69], where the textual explanations to be shown to the user were determined based on the user's preferences and constructed from pre-defined templates.

Providing more information about a recommended item, e.g., in the form of pros and cons, is a typical approach when providing explanations. Generating such item descriptions in a user-tailored way in the context of CRS was proposed in Reference [43] and Reference [150]. In such approaches, the users' feedback during the conversation can influence which attributes are mentioned in the item descriptions shown to the user in the recommendation phase. Furthermore, the user preferences can be considered to order the arguments and to help determine which adjectives and adverbs to use in the explanation [43].

In Reference [101], two kinds of explanations were implemented in a CRS for movies. One was simply based on the details of a given movie, whereas the other connects the given user preferences with item features through a graph-based approach to create a personalized explanation. Another graph-based approach following similar ideas was proposed in Reference [97], where a knowledge-augmented dialogue system for open-ended conversations was discussed. In this approach, relevant entities and attributes in a dialogue context were retrieved by walking over a common fact knowledge graph, and the walk path was used to create explanations for a given recommendation. In Reference [109], finally, a human-centered approach was employed. By analyzing a human-human dialogue dataset, the authors identified different social strategies for explaining movie recommendations. They then accommodated the social explanation in a conversational recommender system to improve the users' perception of the quality of the system.

However, for the main task of explaining, we found that little CRS-specific research exists so far, and only a smaller set of the proposed CRS in the literature support such a functionality.

5.1.4 Respond. This category of tasks is relevant in user-driven or mixed-initiative NLP-based CRS, where the user can actively ask questions to the system, actively make preference statements, or issue commands. The system's goal is to properly react to user utterances that do not fall in the above-mentioned categories "Recommend" and "Explain." Two main types of technical approaches can be adopted to respond to such user utterances. One approach—also commonly used in chatbots—is to map the utterances to pre-defined *intents*, such as the ones mentioned in Table 1, e.g., Obtain Details or Restart. The system's answers to these pre-defined intents can be implemented in the system with the help of templates. In the literature, various user utterances are mentioned to which a CRS should be able to respond appropriately. Examples include preference refinements, e.g., "*I like Pulp Fiction, but not Quentin Tarantino*" [101], requests for more information about an item, or a request for the system's judgement regarding a certain item, e.g., "*How about Huawei P9 ?*" [164].

An alternative technical approach is to select or generate the system's responses by automatically training a machine learning model from dialogue corpora and other knowledge sources like in "end-to-end" learning systems, e.g., References [27, 63, 64, 75]. In an open-domain dialogue system described in Reference [114], for example, an information retrieval model was used to retrieve an initial set of candidate answers from a Q&A knowledge base (an online customer service chat log). Then, an attentive sequence-to-sequence model was used to rank the candidate answers to determine answers with scores that are higher than a pre-defined threshold. If no existing answer was considered suitable, then a sequence-to-sequence based model was used to generate the system's response.

Another example for such an approach is described in Reference [105]. In this multi-modal recommender system, one RNN model was used to generate general responses such as greetings or chit-chat, and a knowledge-aware RNN model was trained to answer more specific questions. For instance, when the user asks for style-tip: "*Will T-shirt complement any of these sandals?*", the system may respond with "*Yes, T-shirt will go well with these sandals*" [105].

Finally, some approaches were proposed in the literature to deal with very specific dialogue situations. One example of such a situation is a *conversational breakdown*, where the system is unable to understand the user's input [98]. Possible repair strategies, such as politeness and apology strategies, were examined in the area of human-robot interaction to mitigate the negative impact of such a breakdown [71, 74, 136]. Various repair strategies based on communication theory, e.g., repeating or asking for clarifications, or strategies from explainable machine learning, e.g., explaining which parts of the conversation were not understood, can in principle be applied [6].

5.2 Supporting Tasks

Depending on the system's functionality, a number of additional and supporting computational tasks may be relevant in a CRS.

Natural Language Understanding. In NLP-based CRS, it is essential that the system understands the users' intents behind their utterances, as this is the basis for the selection of an appropriate system action [118]. Two main tasks in this context are *intent detection* and *named entity recognition*, and typical CRS architectures have corresponding components for this task. In principle, intent detection can be seen as a classification task (dialogue act classification), where user utterances are assigned to one or multiple intent categories [137]. Named entity recognition aims at the identification of entities in a given utterance into pre-defined categories such as product names, product attributes, and attribute values [164].

Although intent detection and named entity recognition have been extensively studied in general dialogue systems [137], there are few studies specific to CRS according to our survey, possibly due to the lack of a well-established taxonomy and large-scale annotated recommendation dialogue data. In an early approach [142], manually-defined *recognition grammars* were used to map user utterances to pre-defined dialogue situations, which is comparable to using pre-defined intents as described above in the context of the *Respond* task. An example for a more recent approach can be found in Reference [164]. Here, a natural language understanding component for intent detection, product category detection, and product attribute extraction was implemented in a dialogue system for online shopping. For instance, from the utterance "*recommend me a Huawei phone with 5.2 inch screen*" the system should derive the intent *recommendation*, the product category *cellphone*, as well as the brand and the display size. To solve these tasks, the authors first collected product-related questions from queries posted on a community site, and then extracted intent phrases (e.g., "*want to buy*" and "*how about*") by using two phrase-based algorithms. A multi-class classifier was trained for intent detection of new user questions. As for product category detection, the authors employed a CNN-based approach that took the detected intent into account to identify the category of a mentioned product in a given utterance.

Neural networks were used also in other recent intent and entity recognition approaches [105, 146]. For example, a Multilayer Perceptron was used to predict the probability distribution on a set of pre-defined intent categories in Reference [105]. A sequence-to-sequence model was used in Reference [166] to reframe the user's query (e.g., "*How to protect my iphone screen*") into keywords (e.g., "*iphone screen protector*") that are then used in the recommendation process to identify candidate items.

Another supporting task in some applications is *sentiment analysis*, see, e.g., References [54, 75, 100, 105, 173]. One typical goal in the context of CRS is to understand a user's opinion about a certain item. For example, whenever an item—e.g., a movie—is mentioned in an utterance, the sentiment of the sentence can be used to approximate the user's feelings about the item. This sentiment can then be considered as an item rating, which can subsequently be used for recommending other items using established recommendation techniques.

Specific Recommendation Functionality. Depending on the technical approach to generate recommendations, specific computational subtasks may be helpful or required to support the recommendation process. We will give examples from the context of critiquing-based approaches here. In References [11, 145], for example, the goal is to make "query suggestions" to users, where the term "query" is equivalent to a critique or constraint on the item features. In the mentioned approaches, the query suggestions (or modifications) are based on an extended analysis of the satisfiability of a query (i.e., will the suggested query lead to any results) or dominance relations between possible query suggestions. Generally, such query suggestions can be particularly helpful for users who have difficulties expressing their preferences. In the field of information retrieval, many approaches to query suggestion were proposed to assist users in expressing their information needs, see, e.g., Reference [135] for a more recent example. Limited work, however, exists so far to apply such ideas to the context of CRS.

A related problem in constraint-based CRS is that in some cases the user's expressed preferences lead to the situation that either too many items remain for recommendation or that no item is left. Different approaches were proposed in the literature for query *relaxation*. In Reference [142], for example, a relatively simple strategy was adopted to remove some constraints. More elaborated strategies were proposed in References [56, 94] and Reference [124]. In this latter work [124], the authors also introduce the concept of "query tightening." Here, the idea is to add more constraints on item attributes in case the number of relevant items returned by the system would lead to a choice overload problem. Generally, like for the query suggestion approaches described above, similar query revision approaches (relaxation and tightening) were not explored to a large extent in the context of NLP-based CRS. An exception is the concept for a chatbot presented in Reference [104], where the system tries to first identify the cause of the unsuccessful query and then asks the user to remove some preferences and to rank the item features by importance. Finally, instead of returning an empty result and asking the user to revise the preferences, approaches exist that automatically relax constraints and inform the user, e.g., that "*There are 10 cameras less than 300 euro but their resolution is between 1 and 4 mega-pixels*" [55, 90].

5.3 Discussion

Our analysis shows that a wide range of technical approaches are used in the literature to support the main computational tasks of a CRS. For the problem of computing recommendations, for example, all sorts of approaches—collaborative, content-based, hybrid—can be used within CRS. However, for the main task of explaining, we found that little CRS-specific research exists so far, and only a smaller set of the proposed CRS in the literature support such a functionality.

Another observation is that dialogue management is often sketched as a conceptual architectural component, but it is then implemented either in a rather static way with pre-defined transitions, e.g., see References [86, 172], or done implicitly during the intent recognition and mapping phase or determined by the choices of the preference acquisition strategy, e.g., slot-filling [162]. In some cases, the possible dialogue states are furthermore quite limited, e.g., the system can either decide to ask questions or to provide a recommendation [139]. Technically, in a few cases intent recognition and dialogue flow management are based on commercial tools, e.g., see References [5, 36].

In general, with the growing spread of chatbot applications, several commercial companies such as Google, Microsoft, Facebook and IBM, have released frameworks or public APIs that implement some of the mentioned computational tasks and allow developers to create their own chatbots. These tools include Google's *DialogFlow* system, Facebook's *Wit.ai*, and IBM *Watson Assistant*[5] and provide functionalities such as speech recognition, voice control, the identification of pre-defined intents from natural language utterances, dialogue flow management, response generation to specific intents, and the deployment of applications to commercial platforms. Examples of research works that used these services include [3, 5, 21, 36, 62, 65, 133]. Frameworks for the development of

[5] https://dialogflow.com, https://wit.ai/, https://www.ibm.com/cloud/watson-assistant/.

conversational systems are also provided by Microsoft through its *Bot Framework* and by Amazon for its Alexa assistant and smart speakers. A CRS for the travel domain that uses Amazon Echo smart speakers was, for example, presented in Reference [4]. In general, however, these frameworks and services usually do not implement functionality that is specific to recommendation problems, but are designed to build general-purpose conversational systems.

Besides companies, also some researchers release their NLP-based CRS to the public. Examples include the VoteGoat movie recommender [36] and the ConveRSE framework for chatbot development [54, 101].

6 EVALUATION OF CONVERSATIONAL RECOMMENDERS

In general, recommender systems can be evaluated along various dimensions, using different methodological approaches [131]. First, when a system is evaluated in its context of use, i.e., when it is deployed, we are usually interested mostly in specific *key performance indicators* that measure—through A/B testing—if the system is achieving what it was designed for, e.g., increased sales numbers or user engagement [57]. Second, user studies (lab experiments) typically investigate questions related to the *perceived quality* of a system. Common quality dimensions are the suitability of the recommendations, the perceived transparency of the process or the ease-of-use, see also Reference [113]. Offline experiments, finally, do not involve users in the evaluation, but assess the quality based on *objective metrics*, e.g., the accuracy of predicting heldout ratings in a test set, by measuring the diversity of recommendations, or by computing running times.

The same set of quality dimensions and research methods can also be applied for CRS. However, when comparing algorithm-oriented research and research on conversational systems, we find that the main focus of the evaluations is often a different one. Since CRS are highly interactive systems, questions related to human–computer interaction aspects are more often investigated for these systems. Furthermore, regarding the measurement approaches, CRS evaluations not only focus on task fulfillment, i.e., if a recommendation was suitable or finally accepted, but also on questions related to the efficiency or quality of the conversation itself.

6.1 Overview of Quality Dimensions, Measurements, and Methods

Through our literature review, we identified the following main categories of quality dimensions investigated in CRS:

(1) *Effectiveness of Task Support*: This category refers to the ability of the CRS to support its main task, e.g., to help the users make a decision or find an item of interest.
(2) *Efficiency of Task Support*: In many cases, researchers are also interested to understand how quickly a user finds an item of interest or makes a decision.
(3) *Quality of the Conversation and Usability Aspects*: Analyses in this category focus on the quality of the conversation itself and on the usability (ease-of-use) of the CRS as a whole.
(4) *Effectiveness of Subtask*: A number of studies investigated in our survey focus on certain subtasks like intent or entity recognition.

In each of these dimensions, a number of different measurements are considered in the literature. Task *effectiveness*, for example, can be both measured objectively (through accuracy measures, acceptance or rejection rates) or subjectively (through surveys related to choice satisfaction or perceived recommendation quality). Task *efficiency* is very often measured objectively through the number of required interaction steps and shorter dialogues are usually considered favorable. The *quality of the conversation* is most often analyzed in terms of subjective assessments, e.g., with respect to fluency, understandability, or the quality of the responses. Finally, specific measurements for subtasks include intent recognition rates or the accuracy of the state recognition process.

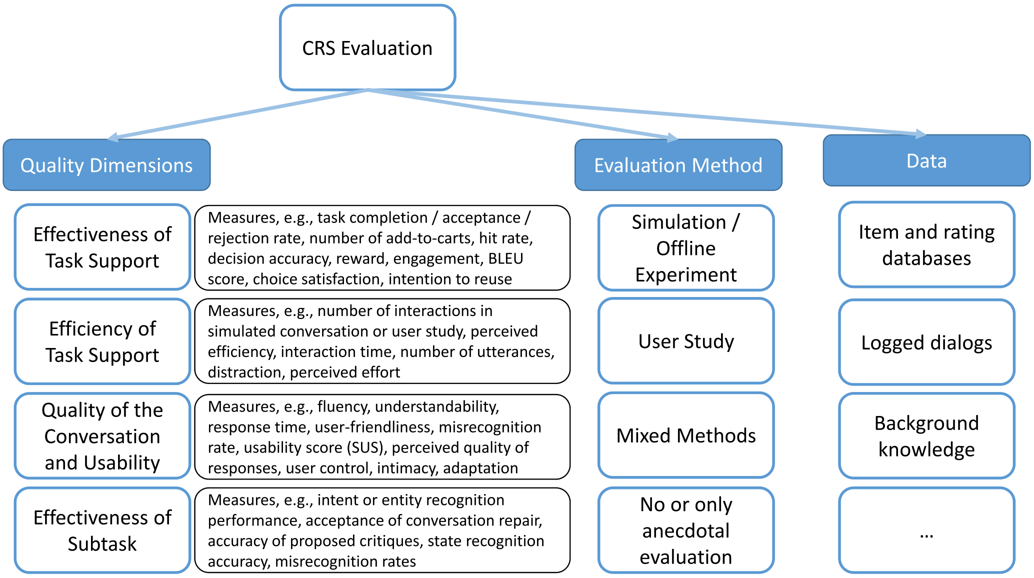
CRS Evaluation
Quality Dimensions
Evaluation Method
Data
Effectiveness of Task Support
Measures, e.g., task completion / acceptance / rejection rate, number of add-to-carts, hit rate, decision accuracy, reward, engagement, BLEU score, choice satisfaction, intention to reuse
Efficiency of Task Support
Measures, e.g., number of interactions in simulated conversation or user study, perceived efficiency, interaction time, number of utterances, distraction, perceived effort
Quality of the Conversation and Usability
Measures, e.g., fluency, understandability, response time, user-friendliness, misrecognition rate, usability score (SUS), perceived quality of responses, user control, intimacy, adaptation
Effectiveness of Subtask
Measures, e.g., intent or entity recognition performance, acceptance of conversation repair, accuracy of proposed critiques, state recognition accuracy, misrecognition rates
Simulation / Offline Experiment
User Study
Mixed Methods
No or only anecdotal evaluation
Item and rating databases
Logged dialogs
Background knowledge
...

Fig. 5. Overview of evaluation dimensions and experiment designs.

From a methodological perspective, we found works that entirely relied on offline experiments, works that relied exclusively on user studies, and studies that combined both offline experiments with user studies. Reports on fielded systems and A/B tests are rare. Examples of such works that discuss deployed systems include [20, 30, 32, 55, 60, 104, 114, 164]. However, the level of detail provided for these tests is often limited, partially informal, or only considers certain aspects like processing times. Finally, we also found works without any evaluation or where the evaluation was mostly qualitative or anecdotal [4, 73, 160].

In the experimental evaluations, all sorts of materials—in particular prototype applications—and datasets were used. As discussed in Section 4, at least an item database is needed. Depending on the technical approach, also additional types of knowledge and data are used, such as logged conversations between humans, explicit dialogue-related knowledge such as supported intents, and so on.

In Figure 5, we provide an overview of the most common evaluation dimensions and evaluation approaches, and give examples for typical measurements and datasets. In the following sections, we will discuss some of the more typical evaluation approaches in more detail.

6.2 Review of Evaluation Approaches

6.2.1 Effectiveness of Task Support. In traditional recommender systems, the most common evaluation approach is to determine—through offline evaluations—how accurate an algorithm is at predicting some known, but withheld user preferences. The underlying assumption is that systems with higher accuracy are more effective, e.g., in helping users find what they need. Objective accuracy measures such as the RMSE or the Hit Rate are sometimes also used to evaluate CRS. However, there are typically no long-term preferences available for conversational systems and the system only learns about the user preferences in the ongoing usage session. Therefore, alternative evaluation protocols are typically applied that rely on simulated users or user studies. Furthermore, researchers sometimes use specific objective metrics besides accuracy, and they also frequently rely on subjective quality assessments from users [27, 75, 139]. The objective and subjective quality measures are discussed below in detail.

Objective Measures. Accuracy measures like Average Precision, the Hit Rate or RMSE were for example used as part of the evaluations in References [18, 29, 101] or Reference [146]. In Reference [29], a framework for interactive preference elicitation was proposed that learns which questions should be asked to users in the cold-start phase. To evaluate different strategies, the authors use real and simulated user profiles and report the average precision of the recommendations after each question-answering round. Similarly, a user simulator was used for the evaluation of a dialogue-based facet-filling recommender system based on deep reinforcement learning and end-to-end memory networks in Reference [146] and Reference [139]. The simulator in Reference [146] was based on real user utterances extracted from a dataset about restaurant reservations [63]. The objective measures included the recommendation accuracy (median of ranking and success rate), as well as the proportion of the simulated users who accepted the recommendations. In Reference [139], the "online" experiments were based on a dataset collected through crowdworkers and the objective measures included Average Reward of the reinforcement learning strategy and the Success Rate (conversion rate), i.e., the fraction of successful dialogues.

The authors of Reference [101] present a domain-independent CRS framework, and they use the Hit Rate to assess the effectiveness of different system components such as the recommendation algorithm or the intent recognizer. To make the measurements, they use the above-mentioned bAbI dataset as a ground truth, where each example contains the user preferences, the recommendation request and the recommended item. A similar evaluation approach based on ground truth information derived from different real-world dialogues and accuracy measures (RMSE, Recall, Hit Rate) was adopted in Reference [27, 64, 75]. In such approaches, the system typically analyzes (positive) mentions of items (movies) in the ongoing natural language dialogue and use these preferences for the prediction task.

The focus of Reference [18] was on implicit feedback in CRS, where this feedback was obtained from non-verbal communication acts. To assess the effectiveness of using such signals, the accuracy of rating predictions by a content-based recommender was evaluated using MAE and RMSE. In their approach, the ground truth for the evaluation was previously collected in a user study. In some ways, this approach is similar to Reference [101] in that the effects of the performance of a side task—here, the interpretation of non-verbal communication acts—on the system's overall recommendation quality are investigated.

Given the possible limitations of pure offline experiments in the context of CRS, user studies are also frequently applied to gauge the effectiveness of a system. In the context of a critiquing-based system [23, 24], for example, *decision accuracy* was objectively measured by the fraction of users who changed their mind when they were presented with all available options after they had previously made their selection with the help of the CRS. In Reference [86], in contrast, the authors used *task completion rates* and *add-to-cart actions* as proxies, which measure how often users had at least one item in their cart and how many items they added on average, respectively.

Subjective Measures. Differently from objective measures that, e.g., record the user's decision behavior when interacting with the system or determine prediction accuracy using simulations, subjective measures assess the user's quality perception of the system. Such measurements can be important because even common accuracy measures are often not predictive of the quality of the recommendations as perceived by the users.[6] In the reviewed literature on CRS, various quality factors were examined that are also commonly used for non-conversational recommenders, e.g., those discussed in the evaluation frameworks in Reference [68] and Reference [113].

[6] See, e.g., References [10, 35, 44, 87, 128].

For the critiquing-based systems discussed in References [23, 24], the authors therefore not only used decision accuracy (as an objective measure) but also assessed factors like *decision confidence*, and *purchase and return intentions. User satisfaction*, either with the system's recommendations or the system as a whole, was additionally investigated in earlier critiquing approaches such as References [112, 125] and in other comparative evaluations [101, 165]. The *perceived recommendation quality* was assessed in the speech-controlled critiquing system in Reference [47], and in Reference [152] the authors looked at *user acceptance rates.* In References [62, 81] and Reference [109], finally, the authors considered several dimensions in their questionnaire like the *match of the recommendations* with the preferences (*interest fit*), the *confidence* that the recommendations will be liked, and *trust.*

6.2.2 Efficiency of Task Support. Traditionally, in particular critiquing-based CRS approaches are often evaluated in terms of the efficiency of the recommendation process. Specifically, one goal of generating dynamic critiques is to minimize the number of required interactions until the user finds the needed item or accepts the recommendation. Such evaluations are often done *offline* with simulated user profiles. One assumption, also in approaches that are not based on critiquing, is that the simulated users act rationally and consistently, i.e., they will not revise their preferences during the process.

Examples of works that measure interaction cycles in critiquing approaches include References [47, 86, 89, 91, 122, 125, 147, 171]. The *number of required interaction stages* was also one of usually multiple evaluation criteria for chatbotlike applications, e.g., References [53, 62, 101, 154], and a shopping decision-aid in Reference [152]. In the context of learning-based systems, the number of dialogue turns in a two-stage interaction model was measured in Reference [139]. The usage of such measures is, however, rather uncommon for natural language, learning-based dialogue systems.

Besides the number of interaction stages, *task completion time* is sometimes used as an alternative or complementary way of objectively measuring efficiency, e.g., in References [62, 86]. In Reference [54], the authors, among other aspects, compared the efficiency of different interaction modes with a chatbot: NLP-based, button-based, and mixed. They measured the number of questions, the interaction time and the time per question in the dialogue. A main outcome of their work was that pure natural language interfaces were leading to less efficient recommendation sessions, in part due to problems of correctly interpreting the natural language utterances.

In the mentioned papers, shorter interaction or task completion times are generally considered favorable. Note however, that in some cases longer sessions are desirable. In particular, longer interaction times might reflect higher user engagement and, as in Reference [62], correspond to a larger number of listened songs in a music application. In Reference [28], the authors compared a voice-based and visual output system and measured the number of options that were explored by the users. In this context, note that the exploration of more items can, depending on the application, both be a sign that the user found more interesting options to inspect and a sign that the user did *not* find something immediately and had to explore more options. In Reference [165], the effects of using a voice interface for a podcast recommender were analyzed. Their results showed that users were slower, explored fewer options, and chose fewer long-tail items, which can be detrimental for discovery.

In some works, finally, subjective measures regarding the efficiency of the process are used, typically as a part of usability assessments. In References [23, 81, 109, 152] and Reference [86] the authors ask the study participants about their perceived *cognitive effort.*

6.2.3 Quality of the Conversation and Usability Aspects. In a number of works, the focus of the evaluation is put on certain aspects of the dialogue quality and on usability aspects regarding the system as a whole. The general *ease-of-use* of the system was, for example examined in References [47, 62, 112, 122]; the more specific concept *task ease* was part of the user questionnaire in Reference [154].

Regarding quality aspects of the conversation itself, various aspects are investigated in the literature. From the perspective of the conversation initiative, the authors of Reference [81] and Reference [109] measured the perceived level of *user control*. Whether or not the desire for control is dependent on personal characteristics was investigated in Reference [62]. In addition to user control, perceived *transparency* was considered as a quality factor in Reference [109]. A common way to establish transparency is through the use of explanations. Questions of how to design explanations for a recommender chatbot were investigated in Reference [108]. The quality factors used in Reference [154] were based on an early framework for evaluating spoken dialogue systems in Reference [78]. They, for example, include *adaptation* (i.e., how fast the system adapts to the user's preferences), *expected behavior* (i.e., how intuitive and natural the dialogue interaction is), or the *entertainment* value. Furthermore, in Reference [109] *coordination*, *mutual attentiveness*, *positivity*, and *rapport* were considered as additional desired factors of a conversation.

Looking closer at the content and linguistic level of the dialogues, many recent proposals based on natural language rely on the *BLEU* [107] score to assess the system's responses, e.g., References [64, 75, 76, 77, 105]. With the help of this score, which was developed in the context of machine translation, one can compare the responses generated by the system with ground-truth responses from real human conversations in an automated way. As an alternative, the *NIST* score can be used, e.g., in Reference [105]. Additional objective linguistic aspects that are measured in the literate include the *lexical diversity* [46], *perplexity* (corresponding to fluency), and *distinct n-gram* (to assess diversity) [27]. In addition to these objective linguistic measures, researchers sometimes consider subjective assessments of the quality of the system responses in their evaluations, e.g., with respect to *fluency*, *appropriateness*, *consistency*, *engagingness*, *relevance*, *informativeness*, and the *overall dialogue quality* and *generation performance* [27, 46, 64, 76, 77, 105, 154].

6.2.4 Effectiveness of Sub Task. In some works, finally, researchers focus on the evaluation of the performance of certain subtasks. Again, such measurements can both be objective or subjective ones. As objective measurements, the *reward* is often computed in approaches that rely on reinforcement learning [86]. In a critiquing system, the number of times a proposed critique was applied was investigated in Reference [122]. In NLP-based systems, in contrast, researchers often evaluate the performance of the entity and intent recognition modules [77, 101]. In the particular multi-modal CRS in Reference [105], Recall was used for assessing the image selection performance. In terms of subjective measures, the *interpretation performance*, i.e., how good the system is in understanding the input, was, for example, considered in Reference [154].

6.3 Discussion

Our review shows that a wide range of different evaluation methodologies and metrics are used to evaluate CRS. In principle, general user-centric evaluation frameworks for recommender systems as proposed in Reference [68] and Reference [113] can be applied for CRS as well. So far, however, while user-centric evaluation is common, these frameworks are not widely used and no standards or extensions to them were proposed in the literature. In terms of objective measurements, typical accuracy measures are used by several researchers. Still, the individual CRS proposals in the literature are quite diverse, e.g., in terms of the application domain, interaction strategy, and background knowledge, and a comparison between existing systems remains challenging.

In NLP-based systems, the BLEU score is widely used for automatic evaluation. However, according to Reference [79], the BLEU score, at least at the sentence level, can correlate poorly with user perceptions, see also Reference [46]. In general, the evaluation of language models is often considered difficult [88] and task-oriented systems like CRS might be even more challenging to assess. These observations therefore suggest that BLEU scores alone cannot inform us well about the quality of the generated system utterances and that in addition subjective evaluations should be applied.

Researchers therefore often resort to offline experiments with simulated users or user studies, where study participants have to accomplish a certain task. In offline studies, often a target (preferred) item is randomly selected, and then a rationally-behaving user is simulated, which interacts with the CRS by answering questions about preferences or by providing feedback on explanations. Such a design however assumes that users *a priori* have fixed preferences towards items or item features. However, in reality, users may also construct or change their preferences during the conversation when they learn about the space of options. Therefore, it is not always fully clear to what extent such simulations reflect real-world situations. In user studies, in contrast, often realistic decision situations are explored and participants have to accomplish tasks like selecting a product in a shop or finding musical tracks for a birthday party. While such studies to some extent remain artificial as usually no real purchase is made, such evaluations seem more realistic than the offline experiments described above. In general, relying solely on offline experimentation seems too limited, except for certain subtasks, given that any CRS is a system that has to support complex user interactions.

Finally, more research seems needed with respect to understand (i) how humans make recommendations to each other in a conversation, and (ii) how users interact with intelligent assistants, e.g., what kind of intelligence they attribute to them and what their expectations are. Some aspects related to these questions are discussed, e.g., in References [29, 65, 108, 165]. With respect to how humans talk with each other, some analyses were done in Reference [13] and Reference [29]. In Reference [13], the authors based their research on insights from the field of *Conversational Analysis*, and correspondingly implement typical real-world conversation patterns, albeit in a somewhat restricted form, in their technical proposal. In general, more work also needs to be done to understand the effects of the quality perception of a system when certain communication patterns like the explanation for a system recommendation are not supported, as it is the case for many investigated systems.

7 OUTLOOK

Our study reveals an increased rise in the area of CRS in the past few years, where the most recent approaches rely on machine learning techniques, in particular deep learning, and natural language based interactions. Despite these advances, a number of research questions remain open, as outlined in the discussion sections throughout the paper. In this final section, we briefly discuss four more general research directions.

One first question is "Which interaction modality supports the user best in a given task?." While voice and written natural language have become more popular recently, more research is required to understand which modality is suited for a given task and situation at hand or if alternative modalities should be offered to the user. An interesting direction of research also lies in the interpretation of non-verbal communication acts by users. Furthermore, entirely voice-based CRS have limitations when it comes to present an entire set of recommendations in one interaction cycle. In such a setting, a *summarization* of a set of recommendations might be needed, as it might in most cases not be meaningful when the CRS reads out several options to the user.

Second, we ask: "What are challenges and requirements in non-standard application environments?" Today, most existing research focuses on interactive web or mobile applications, either with forms and buttons or with natural language input in chatbot applications. Some of the discussed works go beyond such scenarios and consider alternative environments where CRS can be used, e.g., within physical stores, in cars, on kiosk solutions, or as a feature of (humanoid) robots. However, little is known so far about the specific requirements, challenges, and opportunities that come with such application scenarios and regarding the critical factors that determine the adoption and value of such systems. Regarding the usage scenarios, most research works discussed in our survey focus on one-to-one communications. However, there are additional scenarios that are not much explored yet, for example, where the CRS supports group decision processes [1, 103].

A third question is "What can we learn from theories of conversation?," see also Reference [141]. Regarding the *underpinnings* and *adoption factors* of CRS, only very few works are based on concepts and insights from Conversation Analysis, Communication Theory or related fields. In some works, at least certain communication patterns in real-world recommendation dialogues were discussed at a qualitative or anecdotal level. What seems to be mostly missing so far, however, is a clearer understanding of what makes a CRS truly helpful, what users expect from such a system, what makes them fail [95], and which intents we should or must support in a system. Explanations are often considered as a main feature for a convincing dialogue, but these aspects are not explored a lot. In addition, more research is required to understand the mechanisms that increase the adoption of CRS, e.g., by increasing the user's trust and developing intimacy [72], or by adapting the communication style (e.g., with respect to the initiative and language) to the individual user.

Finally, from a *technical* and *methodological* perspective, we ask: "How far do we get with pure end-to-end learning approaches, i.e., by creating systems where, besides the item database, only a corpus of past conversations serves as input. Tremendous advances were made in NLP technology in recent years, but it stands to question if today's learning-based CRS are actually useful, see Reference [59]. In part, the problem of assessing this aspect is tied to how we evaluate such systems. Computational metrics like BLEU can only answer certain aspects of the question. But also the human evaluations in the reviewed papers are sometimes not too insightful, in particular when a newly proposed system is evaluated relative to a previous system by a few human judges. We therefore should revisit our evaluation practice and also investigate what users actually expect from a CRS, how tolerant they are with respect to misunderstandings or poor recommendations, how we can influence these expectations, and how useful the systems are considered on an absolute scale. Technically, combining learning techniques with other sorts of structured knowledge seems to be key to more usable, reliable and also predictable conversational recommender systems in the future.

REFERENCES

[1] Jesús Omar Álvarez Márquez and Jürgen Ziegler. 2016. Hootle+: A group recommender system supporting preference negotiation. In *Collaboration and Technology (CRIWG'16)*. 151–166.

[2] Elisabeth André and Catherine Pelachaud. 2010. Interacting with embodied conversational agents. In *Speech Technology: Theory and Applications*. Springer US, 123–149.

[3] Prashanti Angara, Miguel Jiménez, Kirti Agarwal, Harshit Jain, Roshni Jain, Ulrike Stege, Sudhakar Ganti, Hausi A. Müller, and Joanna W. Ng. 2017. Foodie Fooderson: A conversational agent for the smart kitchen. In *CASCON'17*. 247–253.

[4] A. Argal, S. Gupta, A. Modi, P. Pandey, S. Shim, and C. Choo. 2018. Intelligent travel chatbot for predictive recommendation in Echo platform. In *CCWC'18*. 176–183.

[5] D. Arteaga, J. Arenas, F. Paz, M. Tupia, and M. Bruzza. 2019. Design of information system architecture for the recommendation of tourist sites in the city of Manta, Ecuador through a chatbot. In *CISTI'19*. 1–6.

[6] Zahra Ashktorab, Mohit Jain, Q. Vera Liao, and Justin D. Weisz. 2019. Resilient chatbots: Repair strategy preferences for conversational breakdowns. In *CHI'19*. 254.

[7] Olga Averjanova, Francesco Ricci, and Quang Nhat Nguyen. 2008. Map-based interaction with a conversational mobile recommender system. In *UBICOMM'08*. 212–218.

[8] Roland Bader, Oliver Siegmund, and Wolfgang Woerndl. 2011. A study on user acceptance of proactive in-vehicle recommender systems. In *AutomotiveUI'11*. 47–54.

[9] Tilman Becker, Nate Blaylock, Ciprian Gerstenberger, Ivana Kruijff-Korbayová, Andreas Korthauer, Manfred Pinkal, Michael Pitz, Peter Poller, and Jan Schehl. 2006. Natural and intuitive multimodal dialogue for in-car applications: The SAMMIE system. In *ECAI'06*. 612–616.

[10] Jöran Beel and Stefan Langer. 2015. A comparison of offline evaluations, online evaluations, and user studies in the context of research-paper recommender systems. In *TPDL'15*. 153–168.

[11] Henry Blanco and Francesco Ricci. 2013. Acquiring user profiles from implicit feedback in a conversational recommender system. In *RecSys'13*. 307–310.

[12] Daniel G. Bobrow, Ronald M. Kaplan, Martin Kay, Donald A. Norman, Henry S. Thompson, and Terry Winograd. 1977. GUS, a frame-driven dialog system. *Artif. Intell.* 8, 2 (1977), 155–173.

[13] Derek G. Bridge. 2002. Towards conversational recommender systems: A dialogue grammar approach. In *ECCBR'02*. 9–22.

[14] Robin Burke. 1999. The Wasabi personal shopper: A case-based recommender system. In *AAAI'99*. 844–849.

[15] Robin D. Burke, Kristian J. Hammond, and Benjamin C. Young. 1997. The FindMe approach to assisted browsing. *IEEE Expert* 12, 4 (1997), 32–40.

[16] Wanling Cai and Li Chen. 2019. Towards a taxonomy of user feedback intents for conversational recommendations. In *RecSys'19 Late-Breaking Results*. 572–573.

[17] Giuseppe Carenini, Jocelyin Smith, and David Poole. 2003. Towards more conversational and collaborative recommender systems. In *IUI'03*. 12–18.

[18] Berardina De Carolis, Marco de Gemmis, Pasquale Lops, and Giuseppe Palestra. 2017. Recognizing users feedback from non-verbal communicative acts in conversational recommender systems. *Pattern Recogn. Lett.* 99, C (2017), 87–95.

[19] Justine Cassell. 2001. Embodied conversational agents: Representation and intelligence in user interfaces. *AI Mag.* 22, 4 (2001), 67–83.

[20] Sapna Ria Chakraborty, M. Anagha, Kartikeya Vats, Khyati Baradia, Tanveer Khan, Sandipan Sarkar, and Sujoy Roychowdhury. 2019. Recommendence and fashionsence: Online fashion advisor for offline experience. In *CoDS-COMAD'19*.

[21] A. A. Chandrashekara, R. K. M. Talluri, S. S. Sivarathri, R. Mitra, P. Calyam, K. Kee, and S. Nair. 2018. Fuzzy-based conversational recommender for data-intensive science gateway applications. In *BigData'18*. 4870–4875.

[22] Fang Chen, Ing-Marie Jonsson, Jessica Villing, and Staffan Larsson. 2010. Application of speech technology in vehicles. In *Speech Technology: Theory and Applications*. Springer, 195–219.

[23] Li Chen and Pearl Pu. 2006. Evaluating critiquing-based recommender agents. In *AAAI'06*. 157–162.

[24] Li Chen and Pearl Pu. 2007. Hybrid critiquing-based recommender systems. In *IUI'07*. 22–31.

[25] Li Chen and Pearl Pu. 2007. Preference-based organization interfaces: Aiding user critiques in recommender systems. In *UMAP'07*. 77–86.

[26] Li Chen and Pearl Pu. 2012. Critiquing-based recommenders: Survey and emerging trends. *User Model. User-Adapt. Interact.* 22, 1-2 (2012), 125–150.

[27] Qibin Chen, Junyang Lin, Yichang Zhang, Ming Ding, Yukuo Cen, Hongxia Yang, and Jie Tang. 2019. Towards knowledge-based recommender dialog system. In *EMNLP-IJCNLP'19*. 1803–1813.

[28] Wen-Kuo Chen, Heng-Chiang Huang, and Seng-cho Timothy Chou. 2008. Understanding consumer recommendation behavior in a mobile phone service context. In *ECIS'08*. 1022–1033.

[29] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards conversational recommender systems. In *KDD'16*. 815–824.

[30] Konstantina Christakopoulou, Alex Beutel, Rui Li, Sagar Jain, and Ed H. Chi. 2018. Q&R: A two-stage approach toward interactive recommendation. In *KDD'18*. 139–148.

[31] F. Clarizia, F. Colace, M. Lombardi, and F. Pascale. 2018. A context aware recommender system for digital storytelling. In *AINA'18*. 542–549.

[32] Francesco Colace, Massimo De Santo, Francesco Pascale, Saverio Lemma, and Marco Lombardi. 2017. BotWheels: A Petri net based chatbot for recommending tires. In *DATA'17*. 350–358.

[33] David Contreras, Maria Salamó, Inmaculada Rodríguez, and Anna Puig. 2015. A 3D visual interface for critiquing-based recommenders: Architecture and interaction. *Int. J. Interact. Media Artif. Intell.* 3 (2015), 7–15.

[34] D. Contreras, M. Salamo, I. Rodriguez, and A. Puig. 2018. Shopping decisions made in a virtual world: Defining a state-based model of collaborative and conversational user-recommender interactions. *IEEE Consum. Electr. Mag.* 7, 4 (2018), 260–35.

[35] Paolo Cremonesi, Franca Garzotto, and Roberto Turrin. 2012. Investigating the persuasion potential of recommender systems from a quality perspective: An empirical study. *Trans. Interact. Intell. Syst.* 2, 2 (2012), 1–41.

[36] Jeffrey Dalton, Victor Ajayi, and Richard Main. 2018. Vote Goat: Conversational movie recommendation. In *SIGIR'18*. 1285–1288.

[37] Berardina De Carolis, Marco de Gemmis, and Pasquale Lops. 2015. A multimodal framework for recognizing emotional feedback in conversational recommender systems. In *EMPIRE Workshop at ACM RecSys*. 11–18.

[38] Doris M. Dehn and Susanne van Mulken. 2000. The impact of animated interface agents: A review of empirical research. *Int. J. Hum.-Comput. Stud.* 52, 1 (2000), 1–22.

[39] Linus W. Dietz, Saadi Myftija, and Wolfgang Wörndl. 2019. Designing a conversational travel recommender system based on data-driven destination characterization. In *ACM RecSys Workshop on Recommenders in Tourism*. 17–21.

[40] 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 *ICLR'16*.

[41] Alexander Felfernig, Gerhard Friedrich, Dietmar Jannach, and Markus Zanker. 2006. An integrated environment for the development of knowledge-based recommender applications. *Int. Electr. Commerce* 11, 2 (2006), 11–34.

[42] Alexander Felfernig, Gerhard Friedrich, Dietmar Jannach, and Markus Zanker. 2015. Constraint-based recommender systems. In *Recommender Systems Handbook*. Springer, 161–190.

[43] Mary Ellen Foster and Jon Oberlander. 2010. User preferences can drive facial expressions: Evaluating an embodied conversational agent in a recommender dialogue system. *User Model. User-Adapt. Interact.* 20, 4 (2010), 341–381.

[44] Florent Garcin, Boi Faltings, Olivier Donatsch, Ayar Alazzawi, Christophe Bruttin, and Amr Huber. 2014. Offline and online evaluation of news recommender systems at Swissinfo.ch. In *RecSys'14*.

[45] Fatih Gedikli, Dietmar Jannach, and Mouzhi Ge. 2014. How should I explain? A comparison of different explanation types for recommender systems. *Int. J. Hum.-Comput. Stud.* 72, 4 (2014), 367–382.

[46] Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. 2018. A knowledge-grounded neural conversation model. In *AAAI'18*. 5110–5117.

[47] Peter Grasch, Alexander Felfernig, and Florian Reinfrank. 2013. ReComment: Towards critiquing-based recommendation with speech interaction. In *RecSys'13*. 157–164.

[48] Claudio Greco, Alessandro Suglia, Pierpaolo Basile, and Giovanni Semeraro. 2017. Converse-Et-Impera: Exploiting deep learning and hierarchical reinforcement learning for conversational recommender systems. In *AI*IA 2017*. 372–386.

[49] Kristian J. Hammond, Robin Burke, and Kathryn Schmitt. 1994. Case-based approach to knowledge navigation. In *AAAI'94*.

[50] Negar Hariri, Bamshad Mobasher, and Robin Burke. 2014. Context adaptation in interactive recommender systems. In *RecSys'14*. 41–48.

[51] Jonathan L. Herlocker, Joseph A. Konstan, and John Riedl. 2000. Explaining collaborative filtering recommendations. In *CSCW'00*. 241–250.

[52] Zeng-Wei Hong, Rui-Tang Huang, Kai-Yi Chin, Chia-Chi Yen, and Jim-Min Lin. 2010. An interactive agent system for supporting knowledge-based recommendation: A case study on an e-novel recommender system. In *ICUIMC'10*. 53:1–53:8.

[53] Yuichiro Ikemoto, Varit Asawavetvutt, Kazuhiro Kuwabara, and Hung-Hsuan Huang. 2019. Tuning a conversation strategy for interactive recommendations in a chatbot setting. *J. Inf. Telecommun.* 3, 2 (2019), 180–195.

[54] Andrea Iovine, Fedelucio Narducci, and Giovanni Semeraro. 2020. Conversational recommender systems and natural language: A study through the ConveRSE framework. *Decis. Supp. Syst.* 131 (2020), 113250–113260.

[55] Dietmar Jannach. 2004. ADVISOR SUITE—A knowledge-based sales advisory system. In *ECAI'04*. 720–724.

[56] Dietmar Jannach. 2006. Finding preferred query relaxations in content-based recommenders. In *IS'06*. 355–360.

[57] Dietmar Jannach and Michael Jugovac. 2019. Measuring the business value of recommender systems. *ACM Trans. Manage. Inf. Syst.* 10, 4 (2019), 1–23.

[58] Dietmar Jannach and Gerold Kreutler. 2007. Rapid development of knowledge-based conversational recommender applications with advisor suite. *J. Web Eng.* 6, 2 (Jun. 2007), 165–192.

[59] Dietmar Jannach and Ahtsham Manzoor. 2020. End-to-end learning for conversational recommendation: A long way to go? In *IntRS Workshop at ACM RecSys 2020*. Online.

[60] Dietmar Jannach, Markus Zanker, Markus Jessenitschnig, and Oskar Seidler. 2007. Developing a conversational travel advisor with ADVISOR SUITE. In *ENTER'07*. 43–52.

[61] Dietmar Jannach, Sidra Naveed, and Michael Jugovac. 2016. User control in recommender systems: Overview and interaction challenges. In *EC-Web'16*.

[62] Yucheng Jin, Wanling Cai, Li Chen, Nyi Nyi Htun, and Katrien Verbert. 2019. MusicBot: Evaluating critiquing-based music recommenders with conversational interaction. In *CIKM'19*. 951–960.

[63] Chaitanya K. Joshi, Fei Mi, and Boi Faltings. 2017. Personalization in goal-oriented dialog. In *NeurIPS'17 Workshop on Conversational AI*.

[64] 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 *EMNLP-IJCNLP'19*. 1951–1961.

[65] Jie Kang, Kyle Condiff, Shuo Chang, Joseph A. Konstan, Loren Terveen, and F. Maxwell Harper. 2017. Understanding how people use natural language to ask for recommendations. In *RecSys'17*. 229–237.

[66] Epaminondas Kapetanios, Doina Tatar, and Christian Sacarea. 2013. *Natural Language Processing: Semantic Aspects*. CRC Press.

[67] Ralph L. Keeney and Howard Raiffa. 1993. *Decisions with Multiple Objectives: Preferences and Value Trade-Offs*. Cambridge UP.

[68] B. P. Knijnenburg, M. C. Willemsen, Z. Gantner, H. Soncu, and C. Newell. 2012. Explaining the user experience of recommender systems. *User Model. User-Adapt. Interact.* 22, 4 (2012), 441–504.

[69] Béatrice Lamche, Ugur Adigüzel, and Wolfgang Wörndl. 2014. Interactive explanations in mobile shopping recommender systems. In *RecSys'19 IntRS Workshop*. 14–21.

[70] Hanbit Lee, Yeonchan Ahn, Haejun Lee, Seungdo Ha, and Sang-goo Lee. 2016. Quote recommendation in dialogue using deep neural network. In *SIGIR'16*. 957–960.

[71] Min Kyung Lee, Sara Kielser, Jodi Forlizzi, Siddhartha Srinivasa, and Paul Rybski. 2010. Gracefully mitigating breakdowns in robotic services. In *HRI'10*. 203–210.

[72] SeoYoung Lee and Junho Choi. 2017. Enhancing user experience with conversational agent for movie recommendation: Effects of self-disclosure and reciprocity. *Int. J. Hum.-Comput. Stud.* 103 (2017), 95–105.

[73] Sunhwan Lee, Robert J. Moore, Guang-Jie Ren, Raphael Arar, and Shun Jiang. 2018. Making personalized recommendation through conversation: Architecture design and recommendation methods. In *AAAI'18*. 727–730.

[74] Yeoreum Lee, Jae-eul Bae, Sona Kwak, and Myungsuk Kim. 2011. The effect of politeness strategy on human-robot collaborative interaction on malfunction of robot vacuum cleaner. In *RSS Workshop on HRI*.

[75] Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards deep conversational recommendations. In *NIPS'18*. 9725–9735.

[76] Xiujun Li, Yun-Nung Chen, Lihong Li, Jianfeng Gao, and Asli Celikyilmaz. 2017. End-to-end task-completion neural dialogue systems. In *IJCNLP'17*.

[77] Lizi Liao, Ryuichi Takanobu, Yunshan Ma, Xun Yang, Minlie Huang, and Tat-Seng Chua. 2019. Deep conversational recommender in travel. arXiv abs/1907.00710. Retrieved from https://arxiv.org/abs/1907.00710.

[78] Diane J. Litman and Shimei Pan. 1999. Empirically evaluating an adaptable spoken dialogue system. In *UM'99*. 55–64.

[79] 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 *EMNLP'16*. 2122–2132.

[80] Jingjing Liu, Stephanie Seneff, and Victor Zue. 2010. Dialogue-oriented review summary generation for spoken dialogue recommendation systems. In *ACL'10*. 64–72.

[81] Benedikt Loepp, Tim Hussein, and Jürgen Ziegler. 2014. Choice-based preference elicitation for collaborative filtering recommender systems. In *CHI'14*. 3085–3094.

[82] Stanley Loh, Daniel Lichtnow, Adriana Justin Cerveira Kampff, and Jose Palazzo Moreira de Oliveira. 2010. Recommendation of complementary material during chat discussions. *Knowl. Manage. E-Learn.* 2, 4 (2010).

[83] Juergen Luettin, Susanne Rothermel, and Mark Andrew. 2019. Future of in-vehicle recommendation Systems @ Bosch. In *RecSys'19*. 524.

[84] Kai Luo, Scott Sanner, Ga Wu, Hanze Li, and Hojin Yang. 2020. Latent linear critiquing for conversational recommender systems. In *WWW'20*. 2535–2541.

[85] Tariq Mahmood and Francesco Ricci. 2007. Learning and adaptivity in interactive recommender systems. In *ICEC'07*. 75–84.

[86] Tariq Mahmood and Francesco Ricci. 2009. Improving recommender systems with adaptive conversational strategies. In *HT'09*. 73–82.

[87] Andrii Maksai, Florent Garcin, and Boi Faltings. 2015. Predicting online performance of news recommender systems through richer evaluation metrics. In *RecSys'15*. 179–186.

[88] Gary Marcus. 2020. GPT-2 and the Nature of Intelligence. Retrieved from https://thegradient.pub/gpt2-and-the-nature-of-intelligence/.

[89] Kevin McCarthy, James Reilly, Lorraine McGinty, and Barry Smyth. 2004. On the dynamic generation of compound critiques in conversational recommender systems. In *AH'04*. 176–184.

[90] Kevin McCarthy, James Reilly, Lorraine McGinty, and Barry Smyth. 2004. Thinking positively-explanatory feedback for conversational recommender systems. In *ECCBR'04*. 115–124.

[91] Kevin McCarthy, Maria Salamó, Lorcan Coyle, Lorraine McGinty, Barry Smyth, and Paddy Nixon. 2006. Group recommender systems: A critiquing based approach. In *IUI'06*. 267–269.

[92] Kevin McCarthy, Yasser Salem, and Barry Smyth. 2010. Experience-based critiquing: Reusing critiquing experiences to improve conversational recommendation. In *ICCBR'10*. 480–494.

[93] Lorraine McGinty and Barry Smyth. 2003. On the role of diversity in conversational recommender systems. In *ICCBR'03*. 276–290.

[94] David McSherry. 2004. Incremental relaxation of unsuccessful queries. In *ECCBR'04*. 331–345.

[95] Mohammed Slim Ben Mimoun, Ingrid Poncin, and Marion Garnier. 2012. Case study–Embodied virtual agents: An analysis on reasons for failure. *J. Retail. Cons. Serv.* 19, 6 (2012), 605–612.

[96] Nader Mirzadeh, Francesco Ricci, and Mukesh Bansal. 2005. Feature selection methods for conversational recommender systems. In *EEE'05*. 772–777.

[97] Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. 2019. OpenDialKG: Explainable conversational reasoning with attention-based walks over knowledge graphs. In *ACL'19*. 845–854.

[98] Chelsea Myers, Anushay Furqan, Jessica Nebolsky, Karina Caro, and Jichen Zhu. 2018. Patterns for how users overcome obstacles in voice user interfaces. In *CHI'18*.

[99] David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Linguist Investig.* 30, 1 (2007), 3–26.

[100] Fedelucio Narducci, Pierpaolo Basile, Andrea Iovine, Marco de Gemmis, Pasquale Lops, and Giovanni Semeraro. 2018. A domain-independent framework for building conversational recommender systems. In *RecSys'18 KaRS Workshop*. 29–34.

[101] Fedelucio Narducci, Marco de Gemmis, Pasquale Lops, and Giovanni Semeraro. 2018. Improving the user experience with a conversational recommender system. In *AI*IA'18*. 528–538.

[102] Fedelucio Narducci, Pierpaolo Basile, Marco de Gemmis, Pasquale Lops, and Giovanni Semeraro. 2019. An investigation on the user interaction modes of conversational recommender systems for the music domain. *UMUAI* 30 (2019), 251–284.

[103] Thuy Ngoc Nguyen and Francesco Ricci. 2017. A chat-based group recommender system for tourism. In *ENTER'17*. 17–30.

[104] Iulia Nica, Oliver A Tazl, and Franz Wotawa. 2018. Chatbot-based tourist recommendations using model-based reasoning. In *Configuration Workshop'18*. 25–30.

[105] Liqiang Nie, Wenjie Wang, Richang Hong, Meng Wang, and Qi Tian. 2019. Multimodal dialog system: Generating responses via adaptive decoders. In *MM'19*. 1098–1106.

[106] Ingrid Nunes and Dietmar Jannach. 2017. A systematic review and taxonomy of explanations in decision support and recommender systems. *User Model. User-Adapt. Interact.* 27, 3–5 (Dec. 2017), 393–444.

[107] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *ACL'02*. 311–318.

[108] Sunjeong Park and Lim. Youn-kyung. 2019. Design considerations for explanations made by a recommender chatbot. In *IASDR'19*.

[109] Florian Pecune, Shruti Murali, Vivian Tsai, Yoichi Matsuyama, and Justine Cassell. 2019. A model of social explanations for a conversational movie recommendation system. In *HAI'19*. 135–143.

[110] Pearl Pu and Li Chen. 2006. Trust building with explanation interfaces. In *IUI'06*. 93–100.

[111] Pearl Pu, Paolo Viappiani, and Boi Faltings. 2006. Increasing user decision accuracy using suggestions. In *CHI'06*. 121–130.

[112] Pearl Pu, Maoan Zhou, and Sylvain Castagnos. 2009. Critiquing recommenders for public taste products. In *RecSys'09*. 249–252.

[113] Pearl Pu, Li Chen, and Rong Hu. 2011. A user-centric evaluation framework for recommender systems. In *RecSys'11*. 157–164.

[114] Minghui Qiu, Feng-Lin Li, Siyu Wang, Xing Gao, Yan Chen, Weipeng Zhao, Haiqing Chen, Jun Huang, and Wei Chu. 2017. Alime chat: A sequence to sequence and rerank based chatbot engine. In *ACL'17*. 498–503.

[115] Filip Radlinski and Nick Craswell. 2017. A theoretical framework for conversational search. In *CHIIR'17*. 117–126.

[116] Filip Radlinski, Krisztian Balog, Bill Byrne, and Karthik Krishnamoorthi. 2019. Coached conversational preference elicitation: A case study in understanding movie preferences. In *SIGDIAL'19*. 353–360.

[117] Dimitrios Rafailidis and Yannis Manolopoulos. 2019. Can virtual assistants produce recommendations? In *WIMS'19*.

[118] Dimitrios Rafailidis and Yannis Manolopoulos. 2019. The technological gap between virtual assistants and recommendation systems. arXiv abs/1901.00431. Retrieved from https://arxiv.orb/abs/1901.00431.

[119] Arpit Rana and Derek Bridge. 2020. Navigation-by-preference: A new conversational recommender with preference-based feedback. In *IUI'20*. 155–165.

[120] Al Mamunur Rashid, Istvan Albert, Dan Cosley, Shyong K. Lam, Sean M. McNee, Joseph A. Konstan, and John Riedl. 2002. Getting to know you: Learning new user preferences in recommender systems. In *IUI'02*. 127–134.

[121] James Reilly, Kevin McCarthy, Lorraine McGinty, and Barry Smyth. 2004. Dynamic critiquing. In *ECCBR 04*. 763–777.

[122] James Reilly, Jiyong Zhang, Lorraine McGinty, Pearl Pu, and Barry Smyth. 2007. A comparison of two compound critiquing systems. In *IUI'07*. 317–320.

[123] Francesco Ricci and Quang Nhat Nguyen. 2007. Acquiring and revising preferences in a critique-based mobile recommender system. *Intell. Syst.* 22, 3 (2007), 22–29.

[124] Francesco Ricci, Adriano Venturini, Dario Cavada, Nader Mirzadeh, Dennis Blaas, and Marisa Nones. 2003. Product recommendation with interactive query management and twofold similarity. In *ICCBR'03*. 479–493.

[125] Francesco Ricci, Quang Nhat Nguyen, and Olga Averjanova. 2010. Exploiting a map-based interface in conversational recommender systems for mobile travelers. In *Tourism Informatics*. IGI, 73–79.

[126] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor. 2015. *Recommender Systems Handbook (2nd ed.)*. Springer-Verlag.

[127] Elaine Rich. 1979. User modeling via stereotypes. *Cogn. Sci.* 3, 4 (1979).

[128] Marco Rossetti, Fabio Stella, and Markus Zanker. 2016. Contrasting offline and online results when evaluating recommendation algorithms. In *RecSys'16*. 31–34.

[129] Amrita Saha, Mitesh M. Khapra, and Karthik Sankaranarayanan. 2018. Towards building large scale multimodal domain-aware conversation systems. In *AAAI'18*.

[130] Yasser Salem, Jun Hong, and Weiru Liu. 2014. History-guided conversational recommendation. In *WWW'14*. 999–1004.

[131] Guy Shani and Asela Gunawardana. 2011. Evaluating recommendation systems. In *Recommender Systems Handbook*. Springer US, 257–297.

[132] Hideo Shimazu. 2002. ExpertClerk: A conversational case-based reasoning tool for developing salesclerk agents in E-commerce webshops. *Artif. Intell. Rev.* 18, 3-4 (2002), 223–244.

[133] N. Siangchin and T. Samanchuen. 2019. Chatbot implementation for ICD-10 recommendation system. In *ICESI'19*. 1–6.

[134] Barry Smyth, Lorraine McGinty, James Reilly, and Kevin McCarthy. 2004. Compound critiques for conversational recommender systems. In *WI'04*. 145–151.

[135] Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *CIKM'15*. 553–562.

[136] Vasant Srinivasan and Leila Takayama. 2016. Help me please: Robot politeness strategies for soliciting help from humans. In *CHI'16*. 4945–4955.

[137] Andreas Stolcke, Noah Coccaro, Rebecca Bates, Paul Taylor, Carol Van Ess-Dykema, Klaus Ries, Elizabeth Shriberg, Daniel Jurafsky, Rachel Martin, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics* 26, 3 (2000), 339–373.

[138] Mingxuan Sun, Fuxin Li, Joonseok Lee, Ke Zhou, Guy Lebanon, and Hongyuan Zha. 2013. Learning multiple-question decision trees for cold-start recommendation. In *WSDM'13*. 445–454.

[139] Yueming Sun and Yi Zhang. 2018. Conversational recommender system. In *SIGIR'18*. 235–244.

[140] P. R. Telang, A. K. Kalia, M. Vukovic, R. Pandita, and M. P. Singh. 2018. A conceptual framework for engineering chatbots. *IEEE Internet Comput.* 22, 06 (2018), 54–59.

[141] Paul Thomas, Mary Czerwinski, Daniel McDuff, and Nick Craswell. 2020. Theories of conversation for conversational IR. In *CAIR'17*.

[142] Cynthia A. Thompson, Mehmet H. Göker, and Pat Langley. 2004. A personalized system for conversational recommendations. *J. Artif. Intell. Res.* 21, 1 (2004), 393–428.

[143] Nava Tintarev and Judith Masthoff. 2011. Designing and evaluating explanations for recommender systems. In *Recommender Systems Handbook*. Vol. 1. Springer, 479–510.

[144] Frederich N. Tou, Michael D. Williams, Richard Fikes, D. Austin Henderson Jr., and Thomas W. Malone. 1982. RABBIT: An intelligent database assistant. In *AAAI'82*. 314–318.

[145] Walid Trabelsi, Nic Wilson, Derek G. Bridge, and Francesco Ricci. 2010. Comparing approaches to preference dominance for conversational recommenders. In *ICTAI'10*. 113–120.

[146] Daisuke Tsumita and Tomohiro Takagi. 2019. Dialogue based recommender system that flexibly mixes utterances and recommendations. In *WI'19*. 51–58.

[147] Paolo Viappiani and Craig Boutilier. 2009. Regret-based optimal recommendation sets in conversational recommender systems. In *RecSys'11*. 101–108.

[148] Paolo Viappiani, Pearl Pu, and Boi Faltings. 2007. Conversational recommenders with adaptive suggestions. In *RecSys'07*. 89–96.

[149] Jesse Vig, Shilad Sen, and John Riedl. 2011. Navigating the tag genome. In *IUI'11*. 93–102.

[150] M. A. Walker, S. J. Whittaker, A. Stent, P. Maloor, J. Moore, M. Johnston, and G. Vasireddy. 2004. Generation and evaluation of user tailored responses in multimodal dialogue. *Cogn. Sci.* 28, 5 (2004), 811–840.

[151] Richard S. Wallace. 2009. The anatomy of A.L.I.C.E. In *Parsing the Turing Test*. Springer, 181–210.

[152] Weiquan Wang and Izak Benbasat. 2013. Research note—A contingency approach to investigating the effects of user-system interaction modes of online decision aids. *Inf. Syst. Res.* 24, 3 (2013), 861–876.

[153] Pontus Wärnestål. 2005. Modeling a dialogue strategy for personalized movie recommendations. In *IUI'05 Beyond Personalization Workshop*. 77–82.

[154] Pontus Wärnestål. 2005. User evaluation of a conversational recommender system. In *IJCAI'05 Workshop on Knowledge and Reasoning in Practical Dialogue Systems.*

[155] Pontus Wärnestål, Lars Degerstedt, and Arne Jönsson. 2007. Interview and delivery: Dialogue strategies for conversational recommender systems. In *NODALIDA'07*. 199–205.

[156] Pontus Wärnestål, Lars Degerstedt, and Arne Jönsson. 2007. PCQL: A formalism for human-like preference dialogues*. In *IJCAI'07 Workshop on Knowledge and Reasoning in Practical Dialogue Systems.*

[157] Bifan Wei, Jun Liu, Qinghua Zheng, Wei Zhang, Chenchen Wang, and Bei Wu. 2015. DF-Miner: Domain-specific facet mining by leveraging the hyperlink structure of Wikipedia. *Knowl.-Based Syst.* 77 (2015), 80–91.

[158] Joseph Weizenbaum. 1966. ELIZA—Computer program for the study of natural language communication between man and machine. *Commun. ACM* 9, 1 (Jan. 1966), 36–45.

[159] Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *ACL'17*. 438–449.

[160] Dwi H. Widyantoro and Z. K. A. Baizal. 2014. A framework of conversational recommender system based on user functional requirements. In *ICoICT'14*. 160–165.

[161] Joshua Wissbroecker and F. Maxwell Harper. 2018. Early lessons from a voice-only interface for finding movies. In *RecSys'19 Late-Breaking Results*.

[162] Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *ACL'19*.

[163] David Jingjun Xu, Izak Benbasat, and Ronald T. Cenfetelli. 2017. A two-stage model of generating product advice: Proposing and testing the complementarity principle. *J. Manage. Inf. Syst.* 34, 3 (2017), 826–862.

[164] Zhao Yan, Nan Duan, Peng Chen, Ming Zhou, Jianshe Zhou, and Zhoujun Li. 2017. Building task-oriented dialogue systems for online shopping. In *AAAI'17*. 4618–4626.

[165] Longqi Yang, Michael Sobolev, Christina Tsangouri, and Deborah Estrin. 2018. Understanding user interactions with podcast recommendations delivered via voice. In *RecSys'18*. 190–194.

[166] Zi Yin, Keng-hao Chang, and Ruofei Zhang. 2017. DeepProbe: Information directed sequence understanding and chatbot design via recurrent neural networks. In *KDD'17*. 2131–2139.

[167] Tong Yu, Yilin Shen, and Hongxia Jin. 2019. A visual dialog augmented interactive recommender system. In *KDD'19*. 157–165.

[168] Tong Yu, Yilin Shen, Ruiyi Zhang, Xiangyu Zeng, and Hongxia Jin. 2019. Vision-language recommendation via attribute augmented multimodal reinforcement learning. In *MM'19*. 39–47.

[169] Markus Zanker and Markus Jessenitschnig. 2009. Case-studies on exploiting explicit customer requirements in recommender systems. *User Model. User-Adapt. Interact.* 19, 1-2 (2009), 133–166.

[170] Jie Zeng, Yukiko I. Nakano, Takeshi Morita, Ichiro Kobayashi, and Takahira Yamaguchi. 2018. Eliciting user food preferences in terms of taste and texture in spoken dialogue systems. In *MHFI'18*. 1–5.

[171] Jiyong Zhang and Pearl Pu. 2006. A comparative study of compound critique generation in conversational recommender systems. In *AH'02*. 234–243.

[172] Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W. Bruce Croft. 2018. Towards conversational search and recommendation: System ask, user respond. In *CIKM'18*. 177–186.

[173] Guoshuai Zhao, Hao Fu, Ruihua Song, Tetsuya Sakai, Zhongxia Chen, Xing Xie, and Xueming Qian. 2019. Personalized reason generation for explainable song recommendation. *ACM Trans. Intell. Syst. Technol.* 10, 4 (2019), 1–21.

[174] Xiaoxue Zhao, Weinan Zhang, and Jun Wang. 2013. Interactive collaborative filtering. In *CIKM'13*. 1411–1420.

Received March 2020; revised November 2020; accepted February 2021