Beyond Single Items: Exploring User Preferences in Item Sets with the Conversational Playlist Curation Dataset

Arun Tejasvi Chaganty
Google Research
arunchaganty@google.com

Megan Leszczynski*
Stanford University
mleszczy@cs.stanford.edu

Shu Zhang
Google Research
shzhang@google.com

Ravi Ganti
Google Research
gmravi@google.com

Krisztian Balog
Google Research
krisztianb@google.com

Filip Radlinski
Google Research
filiprad@google.com

arXiv:2303.06791v2 [cs.IR] 5 May 2023

ABSTRACT

Users in consumption domains, like music, are often able to more efficiently provide preferences over a set of items (e.g. a playlist or radio) than over single items (e.g. songs). Unfortunately, this is an underexplored area of research, with most existing recommendation systems limited to understanding preferences over single items. Curating an item set exponentiates the search space that recommender systems must consider (all subsets of items!): this motivates conversational approaches—where users explicitly state or refine their preferences and systems elicit preferences in natural language—as an efficient way to understand user needs. We call this task conversational item set curation and present a novel data collection methodology that efficiently collects realistic preferences about item sets in a conversational setting by observing both item-level and set-level feedback. We apply this methodology to music recommendation to build the Conversational Playlist Curation Dataset (CPCD)[1], where we show that it leads raters to express preferences that would not be otherwise expressed. Finally, we propose a wide range of conversational retrieval models as baselines for this task and evaluate them on the dataset.

CCS CONCEPTS

• **Information systems** → *Music retrieval*; **Collaborative search**; **Test collections**; **Recommender systems**; • **Computing methodologies** → *Natural language processing*.

KEYWORDS

conversational recommendation, natural language processing, dataset

---

*Work done while interning at Google Research.

[1]The dataset can be viewed at https://github.com/google-research-datasets/cpcd.

Figure 1: An example conversation from the dataset. In conversational item set curation, users collaborate with the system to iteratively curate an *item set* as opposed to a single item. As a result, user preferences often apply at the set-level (e.g., "a playlist for a sunny day") and center on thematic cohesion (e.g., "Early jazz is my favorite when it comes to flow") and diversity ("More similar artists").

1 INTRODUCTION

In recent years, conversational agents have become increasingly popular for a variety of tasks, including recommendation systems [20, 47]. The conversational paradigm is especially suited for recommendation tasks, because of the convenience of using natural language to express preferences and provide feedback [25, 44, 50, 54]. Most existing conversational recommendation methods focus on recommending individual items [20]. However, there are also scenarios where the user aims to create a set of items from a larger collection—creating a music playlist for a given context, like working out or for a long drive, is a common manifestation of this problem. In this setting, the goal of the conversational agent is to aid the user in the selection of a *set of items* that the user has in

mind. The problem of recommending sets of items in a conversational setting, which we refer to as *conversational item set curation*, is still largely unexplored. We use the term curation to highlight that this is a collaborative and iterative process that requires the conversational agent to elicit and understand the user's preferences in order to effectively aid them. In this work, we propose a framing of this novel problem, create a representative dataset, and present and experimentally evaluate a set of baselines.

The problem of conversational item set curation differs from prior work on entity set retrieval [7, 9, 18, 48, 49] in a number of important ways. First, what constitutes a relevant item for a set is highly subjective; different people would find different music "energetic" or "relaxing" for a given context. The conversational setting allows for natural feedback both on the item and on the set level. Second, the curation of the item set is more iterative and interactive than entity set retrieval, as the conversational agent must engage in a back-and-forth dialogue with the user in order to fully understand their preferences. This requires the development of more sophisticated dialog management techniques, including effective strategies for balancing exploration and exploitation as well as the ability to handle uncertainty and ambiguity in the user's responses. We break down the problem of conversational item set curation into a set of more specific tasks, in order to provide a clear and specific focus for further research in this space. Specifically, (1) *conversational item retrieval* is the task of returning relevant items from a collection based on the conversation history; (2) *conversational item set refinement* identifies which and how many retrieved items to use to update the item set; and (3) *conversational elicitation/explanation generation* is concerned with the generation of questions for eliciting user preferences either via direct questions or indirectly via user feedback provided on the explanations that accompany the system's recommendations.

A prerequisite first step to allow research on the above tasks is the construction of a dataset with the desired characteristics. We present a novel human-to-human data collection methodology that efficiently acquires realistic preferences about item sets that are incrementally constructed, observing both item-level and set-level feedback. Specifically, we perform the data collection within the domain of music recommendation, although we expect it to be applicable to most recommendation domains, where item sets are natural units of consumption (e.g., compiling reading lists or healthy dinner recipes). We implement a high-fidelity multi-modal conversational user interface, where users can listen to songs and leave both item- and set-level feedback. Using this methodology, we build and publicly release the Conversational Playlist Creation Dataset (CPCD), comprising of over 900 dialogues containing with 4,800 utterances that express a wide range of preference statements and are paired with over 22,000 item ratings. See Figure 1 for an example conversation from the dataset.

Finally, we use this dataset to experimentally compare a range of conversational retrieval models that are meant to serve as representative baselines. Specifically, we focus on the first sub-task of conversational item retrieval and propose an evaluation methodology that adapts standard retrieval metrics to the multi-turn conversational setting. We find that, while both sparse and dense retrieval methods are useful for this task, our best performing method is a dual encoder architecture [37] pretrained using unsupervised

contrastive learning [19] and fine-tuned on the CPCD development set. There remains significant headroom for improvement.

In summary, our key contributions are:

(1) We introduce the task of conversational item set curation, a collaborative task where a user interacts with a system to curate a (small) set of items that meet the user's goal from a (large) collection.
(2) We develop a data collection methodology for conversational item set curation.
(3) We apply this methodology in the music domain to collect the Conversational Playlist Curation Dataset, published at https://github.com/google-research-datasets/cpcd.
(4) We present a wide range of conversational retrieval models as baseline approaches and evaluate them on this dataset.

2 RELATED WORK

2.1 Conversational Recommendation Datasets

A key contribution of this work is a new conversational recommendation dataset, the Conversational Playlist Curation Dataset (CPCD). There are several existing conversational recommendation datasets collected by pairing a "user" and "wizard" to talk to each other: in some settings, as with CPCD, the user has an open-ended dialog with the wizard [6, 17, 31, 39]; in others users are asked to follow instructions (e.g., mention a specific preference) in each turn of the dialog [6, 21, 33, 34, 36, 53]. Existing datasets all focus on recommending a single item (typically a movie) to the user; in contrast, our work focuses on recommending a *set of items* (songs in particular). To the best of our knowledge, CPCD is the first such dataset. This distinction leads to several important differences in methodology from prior work: first, we allow users and wizards to preview recommendations so that they can react accordingly;[2] second, we allow users to share explicit item ratings with wizards in addition to their responses, leading to more open-ended feedback that often includes soft attributes [3]; finally, we employ wizards with relevant domain expertise to respond with relevant recommendations. In addition, CPCD provides a larger set of candidate and relevant items, making it suitable to evaluate the recommendation component of conversational recommendation systems. Table 1 summarizes some of the key differences of CPCD with prior work.

An alternative to manually collecting data include synthetic data approaches for conversational recommendation systems (CRSs), e.g., using item rating data [13], textual review data [50], or user logs (e.g., watch history) [53]. Dodge et al. [13] and Zhang et al. [50] generate dialogs using natural language templates, while Zhou et al. [53] retrieve candidate utterances and then use human annotators to rewrite them to be more conversational.

2.2 Conversational Information Seeking

Beyond recommendation, conversational information seeking (CIS) systems help users find information through a sequence of interactions that can include search and question answering [47]. Such systems also seek to enable users to provide direct feedback to improve

---

[2]While Hayati et al. [17] allow users to preview recommended movies at the end of the task prior to rating them, they can't do so during the conversation.

Dataset	CPCD	ReDial [31]	Inspired [17]	DuRecDial [33, 34]	GoRecDial [21]	TaskMaster-2 [6]
Domain(s)	Music	Movies	Movies	Movies	Movies	Music (and others)
# of Dialogs	917	11,348	1,001	10,190	9,125	1,602
# of Turns / Dialog	5.8	9.1	10.7	7.6	9.9	8.9
# of Ratings / Dialog	24.5	4.45	1	1	-	-
# of Candidates	106,736	-	-	10	5	-
Open-ended dialog	✓	✓	✓	✗	✗	✓ (partial)
Item set recommendations	✓	✗	✗	✗	✗	✗
Shared item ratings	✓	✗	✗	✗	✗	✗
Item preview	✓	✗	✓ (post-task)	✗	✗	✓
Domain-Expert Wizards	✓	✗	✗	✗	✗	✗

Table 1: Comparison of conversational recommendation datasets. In contrast to existing datasets, CPCD focuses on item set recommendation in the music setting. To allow users to provide richer feedback and wizards to respond with more relevant results, we allow them to preview and rate recommendations during the conversation, and employ domain experts as wizards.

their results without needing to rely on as much historical interaction data as collaborative filtering-based RSs [23, 24]. As a result, our work builds on existing work in modeling conversational search and conversational question answering [8, 16, 26, 29, 35, 46, 51, 55].

2.3 Entity Retrieval

Finally, item set recommendation is closely related to entity retrieval [1], with analogies to entity list completion or example-augmented search [4, 5, 7, 10–12, 18, 48, 49]. In entity list completion, the input consists of a textual query and a small set of example entities; in conversational item set recommendation, the input consists of the conversation history (textual query) and the current item set (example entities). We follow recent work in dense entity retrieval, and evaluate baselines that use a dual encoder to embed queries and items [15, 28, 30].

3 TASK DEFINITION

Conversational item set curation is a collaborative task where a user interacts with a system to *curate* a *set of items* from a larger collection (Figure 2). We use the term *curate* to emphasize the collaborative and iterative nature of the task: the goal is to identify not just a single item that matches the user's preferences, but a collection of items that may span a range of interests, but must be ultimately cohesive with the theme or goal of the collection. While some themes may be narrow ("collection of Elvis Preseley's greatest hits"), others may be broad ("rock music for a long drive").

In the following, we focus on the music domain where items are songs and the item set is a playlist. The user begins the task by stating an overall theme or goal for their item set in their first utterance X_1, e.g. "I'd like to create a playlist to listen to during Christmas with the family," and the system (possibly proxied by a human wizard) responds with an initial playlist Z_1 and an utterance E_1 that explains its choice and elicits more information from the user. In following turns t, the user critiques their working playlist Z_{t-1} in their response to the system X_t. The system in turn uses the user feedback to retrieve additional songs Y_t and responds with an updated playlist Z_t and response E_t. Note that the system must decide how many results (if any) from Y_t it should include when

updating the playlist Z_t. This process continues for T turns until the user is satisfied with their playlist.

Conversational item set curation (CIsC) can be formalized as the task of going from the user goal X_1 to an item set that meets their goal Z_T. Given a conversation history, $H_t = (X_1, Z_1, E_1, \ldots, X_t)$, a conversational item set curation system must solve three subtasks: first, it must retrieve relevant items Y_t from the item corpus (conversational item retrieval), then it must update the item set Z_t (conversational item set refinement) and finally it must generate a response E_t (conversational elicitation / explanation generation). In this paper, our primary focus will be on conversational item retrieval.

Next, we describe the sub-tasks in further detail:

Conversational Item Retrieval (CIRt). In each turn, the system uses the conversation history H_t to retrieve and rank relevant items Y_t from the item collection C. To provide contextually relevant recommendations to the user, the system must understand refinements or critiques ("Got anything from old Stevie Wonder?" in Figure 2) and soft attributes ("playlist for a sunny day" in Figure 1). Finally, we note that the scope of CIRt limited to preferences stated by the user thus far; eliciting further preferences is handled below. We propose evaluating the output of CIRt, Y_t, against the target item set Z_T using standard information retrieval metrics such as MRR or Hits@k in Section 7.

Conversational Item Set Refinement (CIsR). Given a list of relevant recommendations from CIRt, the task of CIsR is to identify which and how many of these items should be used when updating the item set. As reasonable heuristic would add the top k items retrieved by CIRt, but a better method could take into account the confidence and diversity of the retrieved items. Alternatively, users could be presented all the results from CIRt and be asked to perform item set refinement themselves. CIsR can be evaluated against Z_t using set-inclusion metrics like precision and recall. We leave further modeling of this component to future work.

Conversational Elicitation / Explanation Generation (CEEG). Finally, user requests are often underspecified; to provide better recommendations, systems can elicit preferences from users, e.g., by asking them for a genre or probing for why they liked what they

Figure 2: An overview of conversational item set curation (and its subtasks). Given a conversation history H_t with previous user utterances $X_1 \ldots X_{t-1}$, item sets Z_{t-1} and system responses $E_1 \ldots E_{t-1}$, the system must first retrieve relevant items Y_t (conversational item retrieval), then update the item set Z_t (conversational item set refinement) and finally generate a response to the user E_t (conversational elicitation / explanation generation).

like [39]. In addition, providing an explanation of its recommendation makes the system more scrutable to users and allows them to better steer the system [2]. While the dataset proposed in this paper includes system responses that exhibit elicitation and explanations, we leave modeling this component to future work.

4 METHODOLOGY

Our key contribution is a new dataset, CPCD, to benchmark conversational item set curation systems. Similar to prior work, we use a human-to-human methodology to collect CPCD: two human raters are recruited to play the role of a user and the system ("a wizard"). However, to faithfully simulate conversational item set recommendation in consumption domains such as music presents additional challenges: first, preferences are extremely personal—"pop music" for one person can be wholly different for another; second, they are deeply contextual—"pop music" for a party may be different from "pop music" for doing household chores; third, they are often grounded in the audio content and are hard to understand using only text; and finally they require substantial domain expertise from wizards to provide relevant recommendations.

We address these challenges as follows: we encourage users to personalize the data collection task by asking them to begin each conversation with a specific music-listening scenario in their own lives (e.g., "walking to work"). This allows them to relate to the task and fixes the context for the conversation and their listening preferences. Next, we implement a high-fidelity interface wherein users and wizards can listen to songs during the task. We found that this was particularly helpful for wizards to better understand users' tastes. Additionally, we implement a multi-modal interface where users can provide explicit item-level feedback in the form of likes and dislikes, allowing them to focus on broader set-level feedback in their replies. Finally, in pilot studies, we found that the task required wizards to not only be familiar with music in general, but also with the sub-genres pertinent to the user's interests: wizards

with less relevant expertise took longer to perform the task, and had less engaging conversations that were driven entirely by the user. We address this problem by asking users and wizards to each fill in a survey of their general genre-level preferences, and then matching users with wizards who were most familiar with the users' preferred genres.

4.1 Interface design

A high-fidelity multimodal interface was critical for this complex data collection task. The user-facing interface displays the conversational history and the current playlist as recommended by the system (Figure 3a). Users can sample songs from their working playlist and provide explicit item-level feedback in the form of likes and dislikes. Users end their turn by responding to the system via text, and are allowed to submit the task after completing at least 5 rounds of conversation and having rated at least 15 songs. Before submitting the task, users complete a short survey where they rate the completeness of their playlist, their satisfaction with the playlist, the understanding capabilities of the system, its helpfulness, and their overall experience with the system (Figure 5d).

Wizards are presented a similar interface with an additional element: a music search tool that wizards use to add songs to the user's playlist (Figure 3b). The music search uses the YouTube Search API. We note that the API is limited in its capabilities and tends to return results with a high lexical overlap. To preserve user privacy, the search results were not personalized. Wizards were encouraged to search for related artists or recommendations via Google or YouTube Music.

4.2 Rater Recruitment and Training

We recruited 10 wizards and 110 users who were fluent English speakers and regularly listened to music from a crowdsourcing vendor. To control for cultural differences, we required both users and wizards to reside in the U.S. The wizards were additionally

(a) User interface.

(b) Wizard interface.

Figure 3: Interface used by users and wizards. Users listen to and rate songs in their shared playlist, and respond to the system by communicating how they would like to improve it. Wizards are able to see the user's ratings, search for music through the interface and elicit preferences from users.

(a) Listening contexts.

(b) Genre familiarity.

Figure 4: Summary of pre-task music survey results. (a) Users and wizards indicated where they listened to music to prime them for the task. Most participants listened to music in multiple contexts. (b) Users and wizards also indicated how familiar they were with different genres on a 4 point Likert scale (shown with color gradation from "Not at all familiar" being lightest to "Very familiar" being darkest); this information was used to match users and wizards.

required to have significant music expertise and have performed several music labeling tasks in the past.

Both users and wizards were provided training material in the form of slides. Users were primed with the following description of the task:

> In this study, you will create a *personal* music playlist for a purpose (e.g., "while doing chores" or "when I'm feeling down") by having a conversation with a system.

They were then asked to come up with several scenarios for each conversation in preparation for the task. Users were informed that another person (the wizard) was facilitating the conversation and

encouraged to be descriptive in their requests by telling the system what they liked or didn't like, and why. Users were given instructions on how to use the interface, but were provided no further guidance on what to say. In contrast, wizards went through multiple pilot rounds of training before beginning the task. During the pilot rounds, they were paired amongst themselves, and given feedback on how they should respond to users. During the task, wizards also assisted users with any interface issues (e.g., how to rate items).

Users and wizards were scheduled to interact for a fixed block of time (between 1 and 3 hours), and conducted multiple conversations during this interval. In order to pair users and wizards, they each completed a short music preferences survey asking them to list

Statistic	ExpertPlaylists	CPCD Dev.	Test
# of examples	15,276	450	467
# of tracks in corpus	332,594	106,736	
Avg. # of turns	-	5.8	5.6
Avg. |query text|	106.5	53.8	55
Avg. |response text|	-	45.6	46.0
Avg. |target item sets|	53.6	19	18.3

Table 2: A summary of key statistics for the Conversational Playlist Creation Dataset (CPCD) and ExpertPlaylists, a proprietary collection of expertly curated playlists.

where they listen to music (Figure 4a) and how familiar they are with various genres (Figure 4b). For each user, we ranked wizards based on their overlap with the genres the user expressed familiarity with and tried to pick the top-ranked available wizard for every user subject to scheduling constraints. Based on pilot feedback, we only matched wizards that were at least somewhat familiar with at least one genre that the user indicated a preference for. To avoid priming their conversations, users and wizards were not informed of which genres they matched on. Further personal information about the raters, such as demographics, was not collected as we did not consider it relevant to the task.

4.3 Post-processing

After collecting the data, we took several manual and automated steps to process the data. First, we standardized all white space in the utterances. Often conversations would begin with the user and wizard checking for each other's presence (e.g., by saying "hi"); these turns were merged with subsequent ones so that the first turn included the goal statement (e.g., "Hi, I'd love to create a playlist for classical songs"). After about 20 minutes, wizards would nudge users to proceed to evaluation: these utterances were removed along with any other turns at the end of the conversation where no songs were added. We then manually reviewed the data to identify and remove any turns where users or wizards side-channeled debugging information (e.g., "I don't see any items," "please rate items so I know what you like"). Finally, we scraped and included metadata (i.e., titles, artists and album names) for each song mentioned in the dataset.

5 DATASET ANALYSIS

We now take a deeper look at the CPCD dataset. In total, after filtering and post-processing, the dataset contains 917 conversations that are randomly split into development and test sets; see Table 2. Table 4 includes a few example conversations from the dataset. On average, each conversation includes about 5.7 turns each for the user and wizard, and ends with a final playlist of about 19 songs. Users rate about 4.8 songs in each turn, of which about 20% are negative ratings. In total, the dataset includes over 100k unique songs that serve as the item corpus for retrieval tasks.

Next, we analyze the distribution of preferences expressed by users and elicited by wizards. To do so, we manually annotated 220 turns from 41 conversations. After exploratory analysis, we identified 6 key categories of preferences for users and and 4 for

Type (Freq.%)	Example
Preferences expressed by users	
Artist (46.0%)	My favorite artist is <u>Drake</u>.
Genre (18.9%)	I love <u>old school rap</u>...
Activity (10.7%)	I'd like to create a playlist that I can utilize <u>for a dance party</u>.
Sound (6.6%)	Anything with a <u>long melodic hook</u> is not good for my workouts.
Mood (6.1%)	So Ambitious is too slow paced for workouts as well. Specifically the hook is <u>not upbeat enough</u>.
Others (11.7%)	More gospel music <u>from the late 2000s</u>. I'd love some songs by <u>British artistes</u>.
Preferences elicited by wizards	
Artist (38.7%)	Of course, who are your favorite artists? You got it. How about some J Cole or Kendrick?
Results (25.8%)	Are any of these good? What about these ones?
Genre (21.0%)	Do you like soft rock, Jazz, pop, classical, or country sounds?
Others (14.5%)	Are you trying to focus as you reorganize? Any particular era or country of origin…? Do you prefer the slower ones?

Table 3: Examples of preferences in the dataset and their estimated frequency. The dataset includes rich preference statements: over 30% of user preferences include soft attributes such as activities, sounds and moods. Many artist preferences use artists as a point of reference (e.g, "Lonnie Liston is nice. More similar artists would be good.").

wizards; Table 3 lists these categories with typical examples. While artist and genre based preferences make up a majority of the dataset, over 30% of preferences include soft attributes such as activities, sounds, moods, and more. Additionally, many artist preferences use artists as a point of reference (e.g., "Maybe something from 2pac and Ice Cube?"). Users express preferences in 80% of turns, while wizards elicit preferences 29% of the time. We also observed that only about 5% of preferences in the dataset are negative preferences, e.g. "Those might be too similar, though I do prefer them to <u>not have any lyrics</u>."

Figure 5 shows how the relative frequency of different preferences change across turns.[3] Activity and genre preferences are often expressed early in the conversation, and maintained as part of the conversation context in later turns. Wizards often begin by eliciting the user for their preferences, and thus add fewer songs to the playlist in the first turn (Figure 5c). We note that wizards often elicited users for artists and genres, which were the attributes that worked most reliably with the search tool; we believe this played a role in the wizards' choices.

---

[3] We only show the first 5 turns in Figure 5 as the number of data points sharply drops after 5 turns.

(a) User Preferences.

(b) Wizard Preferences.

(c) Per-turn Item Ratings.

(d) Post-task User Ratings.

Figure 5: (a, b) Preferences expressed by users and elicited by wizards through their conversations. Users often start the conversation with activity and genre preferences, which are maintained as part of context, and then provide additional artist based preferences (often as examples) in later turns. Likewise, wizards begin conversations by eliciting users for artist and genre preferences.

Finally, Figure 5d summarizes the distribution of post-task user ratings. Users were overall satisfied by the experience and expressed that the system understood their preferences well and was very helpful in assisting them on the task. They also expressed that they thought their playlist was mostly complete, and that they would listen to it again. The post-task ratings were weakly correlated with the match scores, with Spearman rank correlation ranging from 0.11 for playlist satisfaction to 0.19 for system helpfulness ($p < 0.05$ in all cases), suggesting marginal benefit beyond ensuring wizards were at least somewhat familiar with genres the user was familiar with.

6 MODELING CONVERSATIONAL ITEM RETRIEVAL

We focus on modeling conversational item retrieval, and leave conversational item set refinement and conversational elicitation and explanation generation to future work. In particular, we consider

three canonical retrieval modeling approaches: sparse retrieval, dense retrieval, and query rewriting.

6.1 Sparse Retrieval: BM25

BM25 is a commonly used sparse, bag-of-words ranking function that measures the relevance of a document or x to a query q. We use BM25 in a sparse retrieval system by treating each item in the retrieval corpus as a candidate document represented in text as "`<title>` by `<artist>` from `<album>`," and returning the top ranked items for each query.[4] We include conversation history in queries by concatenating user queries from previous turns (omitting conversation history led to worse performance). For consistency with the dense retrieval methods below, we use SentencePiece [27] to tokenize both queries and documents.

---

[4]We used the hyperparameters $k_1 = 1.5$ and $b = 0.75$ in our implementation.

User	I want to create a playlist for a girls night with some music from Nicki Minaj.
System	Got you! What do you think about these songs by Nicki?
	✓ Feeling Myself by Nicki Minaj, Beyoncé
	✓ Beez In The Trap by Nicki Minaj, 2 Chainz
	✗ Super Bass by Nicki Minaj
User	Yes that is exactly what I am looking for
System	Perfect! What about these?
	✓ Bartier Cardi by Cardi B, 21 Savage
	✗ Ain't Sh*t by Doja Cat
	✗ Big Energy by Latto
User	I definitely would like to have more Cardi B on this playlist. Not liking the songs by the other artists too much.
System	No worries, here are some songs by Cardi B
	✓ WAP by Cardi B, Megan Thee Stallion
	✓ Thru Your Phone by Cardi B
	✓ Money by Cardi B
User	I want to create a playlist for working out
System	Of course, who are your favorite artists?
User	I enjoy working out to Hip Hop music. Some of my favorite artists are JayZ and Eminem
System	You got it
	✓ The Way I Am by Eminem
	✓ So Ambitious by Jay-Z, Pharrell Williams
	✗ Young Forever by Jay-Z, Mr Hudson
User	The song "Young Forever" is too slow paced for a workout playlist. Great song though.
System	Here's a few uptempo Jay-Z songs for you
	✓ On To The Next One by Jay-Z, Swizz Beatz
	✓ I Got the Keys by DJ Khaled, Jay-Z, Future
	✗ Hey Papi by JAY-Z, Memphis Bleek, Amil
User	So Ambitious is too slow paced for workouts as well. Specifically the hook is not upbeat enough.
System	Here's a few more options for you
	✓ Run This Town by Jay-Z, Kanye West, Rihanna
	✓ GOD DID by DJ Khaled, Rick Ross, Lil Wayne, Jay-Z, John Legend, Fridayy
	✓ D.O.A. (Death Of Auto-Tune) by Jay-Z
User	I'd like to create a playlist that will put me to sleep, maybe some instrumentals
System	Sounds great! Here's a few to start
	✓ Oase by Mo Anando
	✓ Rainbow Roll by Strehlow
	✗ throwback port by mommy, Philanthrope
User	those are pretty songs
System	Here are a few more along a similar vein for your review
	✓ Just Chilling by Grisp
	✓ Life Circle by mr. käfer
	✗ Raindrops Keep Fallin' by Phlocalyst
User	Those might be too similar, though I do prefer them to not have any lyrics
System	Ok, here are some very different instrumentals
	✓ Black Mountain Side (Remaster) by Led Zeppelin
	✗ Electric Worm by Beastie Boys
	✗ Frankenstein by The Edgar Winter Group

Table 4: Examples of conversations in the dataset. The examples showcase the breadth of music interests and listening contexts in the dataset, and how users center thematic cohesiveness and diversity in their preferences. Due to space considerations, only the first three turns and up to three results from each turn are shown.

6.2 Dense Retrieval Methods

We consider a popular and efficient dense retrieval method: a dual encoder [15, 22, 37]. Dual encoders independently embed queries q and items x into normalized dense vectors using a neural network,

and retrieve the most similar items for a query using cosine similarity: $\rho(x; q) = f(q)^{\top} g(x)$, where $f : Q \rightarrow \mathbb{R}^d$ and $g : X \rightarrow \mathbb{R}^d$ are embedding functions for queries and items, respectively.

At turn t, we represent the using the conversation history $H_t = (X_1, Z_1, E_1, \ldots, X_t)$ in text by concatenating user utterances with text representations of the top-k ($k = 3$) songs added to the item set in each turn in reverse chronological order. For example, the query at turn t is:

$$X_t [\text{SEP}] d(\delta Z_{t-1})[\text{SEP}] X_{t-1} \ldots d(\delta Z_1)[\text{SEP}] X_1,$$

where $d(\delta Z_t)$ is a textual representation of the items (same as in BM25) added to the item set in turn t, and [SEP] represents a separator token. We do not include the previous system response (E_t) in the query.

6.2.1 Training and Inference. For training, we use a standard contrastive loss with in-batch negatives: given an example i with query q_i and target item set Z_i, we randomly sample an item from $x_i \in Z_i$ to be a positive target, and take items from other item sets in the batch, x_j where $j \neq i$, to be negative targets. Additionally, we augment the training data to improve robustness as follows: (1) we generate conversations of varying lengths by randomly truncating conversations to its first t turns, and (2) we generate historical item sets of varying lengths by randomly truncating them to their first k items.

6.2.2 Model implementation. We use a shared encoder for f and g initialized either from a pre-trained T5 1.1-Base checkpoint [40] or a T5 1.1-Base checkpoint that has been further pre-trained on retrieval tasks using the Contriever objective [19]. We then fine-tune the encoder on the CPCD development set for 1,000 steps using Adafactor [41] with a batch size of 512 and a constant learning rate of 1e-3 (further fine-tuning led to overfitting). We denote these methods as DE▷CPCD and Contriever▷CPCD respectively.

Additionally, to explore other relevant training data, we also fine-tuned a T5 1.1-Base checkpoint on a proprietary collection of 15,276 expertly curated playlists, ExpertPlaylists, by using its playlist descriptions as queries and the playlists themselves as the target item sets. We denote this system as DE▷ExpertPlaylists.

For inference, we build an index of pre-computed item embeddings using each method's respective encoder. We embed queries as in training, and use nearest neighbor search to return the top-k items for q_t.

6.3 Query rewriting methods

Query rewriting or de-contextualization rewrites a query to incorporate conversation history, and has been widely used in state-of-the-art conversational search systems [32, 42, 45]. Following prior work [32], we fine-tune a T5-based query rewriter on the CANARD dataset [14] and use it to rewrite the queries in CPCD. We adapt two of the above methods, BM25 and DE▷ExpertPlaylists, to use a query rewriter by replacing the conversation history with rewritten queries; the resulting methods are named BM25+QR and DE▷ExpertPlaylists+QR respectively.

Model	Hits@10	Hits@20	Hits@100
Popularity	7.8	14.8	37.9
BM25	19.7	27.4	45.5
BM25+QR	15.5	21.7	34.0
DE▷ExpertPlaylists	13.1	19.6	43.2
DE▷ExpertPlaylists+QR	13.1	20.2	42.5
DE▷CPCD	10.7	16.1	30.1
Contriever▷CPCD	**27.5**	**36.2**	**49.6**

Table 5: Hits@k on CPCD test. DE denotes dual encoder; QR denotes query rewriter. Underlined numbers denote statistical significance compared to the popularity baseline, according to a paired randomization test ($p < 0.05$).

7 EVALUATING RECOMMENDATION PERFORMANCE

We now use CPCD to evaluate the conversational item retrieval methods described in Section 6.

7.1 Experimental Setup

7.1.1 Baselines. We evaluate the following baseline methods: BM25, BM25+QR, DE▷ExpertPlaylists, DE▷ExpertPlaylists+QR, DE▷CPCD, and Contriever▷CPCD. Additionally, we include a popularity baseline that returns the top 100 songs from the development set for every query in the test set.

7.1.2 Evaluation metrics. Evaluating recommendation systems is challenging because there are often missing ratings [13, 43]: we do not know whether an unrated song would be liked or disliked by a user. Standard ranking-based metrics treat items without ratings as not relevant—even though they may actually be relevant—and thus provide a lower bound on the recommendation performance. Following prior work [13, 21, 52], we compare systems using a standard ranking metric, Hits@k, which is 1 if and only if any of the top-k retrieved songs are in the target item set. Unless otherwise stated, we report a macro-averaged Hits@k, by averaging Hits@k across turns within a conversation and then across conversations. We report Hits@k for multiple values of k (10, 20, 100): while smaller values are more representative of realistic scenarios where users only consider a small set of items, Valcarce et al. [43] found that larger values of k have more discriminative power in offline experiments and can help mitigate the missing rating problem.

Evaluating *conversational* recommendation systems poses an additional challenge: how to support evaluation across multiple turns. Different systems may recommend different items in previous turns, leading to divergent histories. To fairly compare across models within a turn, we assume a shared "gold" history and target item set across all models. Additionally, we remove any items in the target item set that also appear in the gold history. When there are additional songs liked in a turn, but not added to the history (e.g., when there is a limit to the number of songs in the history), we keep them in the target item set as leftovers [38], resulting in a larger target item set.

7.2 Main Results

Table 5 compares models on the CPCD test set. As expected, the popularity baseline is among the worst evaluated, and its performance reflects the degree of overlap between the development and test sets: about 26% of songs liked in the development set were also liked in the test set and 38% of the top 100 songs in the development set also appear in top 100 songs of the test set. Next, the sparse retrieval baseline, BM25, does significantly better than the popularity baseline and is the second best method overall. In particular, we find that BM25 scores highly on artist-based queries which are well-suited to sparse bag-of-words based methods. Furthermore, the search tool used by wizards appears to rely on lexical overlap, which may bias the results in favor of sparse methods. We found that using query rewriting either did not improve or hurt performance: we attribute this to the rewriter not generalizing well to the music-seeking queries in CPCD; better query rewriting methods are a promising avenue for future work. Finally, the performance of the dense retrieval methods depended on both the fine-tuning dataset as well as the pretraining strategy: while training on playlist descriptions (DE▷ExpertPlaylists) outperforms the popularity baseline, it still lags far behind BM25, and training solely on the CPCD (DE▷CPCD) does even worse due to insufficient training data. In contrast, training on CPCD from a model pretrained for the retrieval task (Contriever▷CPCD) is the best method we evaluated and significantly outperforms BM25 ($p < 0.05$).

Recommendations on using CPCD for model training. Given its relatively small size, it can be challenging to train a dense retrieval model using CPCD alone (e.g., DE▷CPCD), and recommend using models pretrained on other resources (e.g., Contriever▷CPCD). When using the dataset for training, we provide canonical train and validation splits of the development set, but encourage e.g., k-fold cross-validation to use the data more efficiently.

8 CONCLUSION

We introduced the task of conversational item set curation to capture recommendation scenarios, such as music consumption, where users are looking for a set of items instead of individual ones. In these settings, we imagine users to collaborate with a system to iteratively curate an item set through conversation. Given a conversation history, we subdivided the task into conversational item retrieval (CIRt), conversational item set refinement (CISR) and conversational elicitation and explanation generation (CEEG). To facilitate research in conversational item set curation and its subtasks, we developed an efficient data collection methodology and used it to create the Conversational Playlist Curation Dataset (CPCD). The dataset contains rich preference statements from both users and wizards that are unique to the item set recommendation setting. Finally, we used the data to evaluate a wide range of conversational item retrieval methods. Specifically, our evaluation focused on the CIRt subtask, where we showed that both sparse and dense retrieval methods are useful. Further modeling—e.g., pretraining paradigms that target conversational music recommendation or explicitly modeling audio content—is necessary to improve performance on this task. In future work, we plan to explore using CPCD how to model and evaluate the remaining subtasks, CISR and CEEG, by utilizing the search actions and responses collected from wizards.

REFERENCES

[1] Krisztian Balog. 2018. *Entity-Oriented Search*. The Information Retrieval Series, Vol. 39.

[2] Krisztian Balog, Filip Radlinski, and Shushan Arakelyan. 2019. Transparent, Scrutable and Explainable User Models for Personalized Recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'19)*. 265–274.

[3] Krisztian Balog, Filip Radlinski, and Alexandros Karatzoglou. 2021. On Interpretation and Measurement of Soft Attributes for Recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*. 890–899.

[4] Krisztian Balog, Pavel Serdyukov, and Arjen P. de Vries. 2011. Overview of the TREC 2010 Entity Track. In *Proceedings of the Nineteenth Text REtrieval Conference (TREC '10)*.

[5] Krisztian Balog, Pavel Serdyukov, and Arjen P. de Vries. 2012. Overview of the TREC 2011 Entity Track. In *Proceedings of the Twentieth Text REtrieval Conference (TREC '11)*.

[6] Saravanan Ganesh Amit Dubey Andy Cedilnik Bill Byrne, Karthik Krishnamoorthi and KyuYoung Kim. 2020. Taskmaster-2. https://github.com/google-research-datasets/Taskmaster/tree/master/TM-2-2020

[7] Marc Bron, Krisztian Balog, and Maarten de Rijke. 2013. Example Based Entity search in the Web of Data. In *Proceedings of the 35th European Conference on Advances in Information Retrieval (ECIR '13)*. 392–403.

[8] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards Conversational Recommender Systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (KDD '16)*. 815–824.

[9] Marek Ciglan, Kjetil Nørvåg, and Ladislav Hluchý. 2012. The SemSets Model for Ad-Hoc Semantic List Search. In *Proceedings of the 21st International Conference on World Wide Web (WWW '12)*. 131–140.

[10] Arjen P de Vries, Anne-Marie Vercoustre, James A Thom, Nick Craswell, and Mounia Lalmas. 2008. Overview of the INEX 2007 Entity Ranking Track. In *Proceedings of the 6th Initiative on the Evaluation of XML Retrieval (INEX '07)*. 245–251.

[11] Gianluca Demartini, Arjen P de Vries, Tereza Iofciu, and Jianhan Zhu. 2009. Overview of the INEX 2008 Entity Ranking Track. In *Advances in Focused Retrieval: 7th International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX '08)*. 243–252.

[12] Gianluca Demartini, Tereza Iofciu, and Arjen P. De Vries. 2009. Overview of the INEX 2009 Entity Ranking Track. In *Proceedings of the Focused Retrieval and Evaluation, and 8th International Conference on Initiative for the Evaluation of XML Retrieval (INEX '09)*. 254–264.

[13] 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 *Proceedings of the 4th International Conference on Learning Representations (ICLR '16)*.

[14] Ahmed Elgohary, Denis Peskov, and Jordan Boyd-Graber. 2019. Can You Unpack That? Learning to Rewrite Questions-in-Context. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP '19)*. 5918–5924.

[15] Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. 2019. Learning Dense Representations for Entity Retrieval. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL '19)*. 528–537.

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

[17] Shirley Anugrah Hayati, Dongyeop Kang, Qingxiaoyang Zhu, Weiyan Shi, and Zhou Yu. 2020. INSPIRED: Toward Sociable Recommendation Dialog Systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP '20)*. 8142–8152.

[18] Yeye He and Dong Xin. 2011. SEISA: Set Expansion by Iterative Similarity Aggregation. In *Proceedings of the 20th International Conference on World Wide Web (WWW '11)*. 427–436.

[19] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised Dense Information Retrieval with Contrastive Learning. (Dec. 2021). arXiv:2112.09118 [cs.IR]

[20] Dietmar Jannach, Ahtsham Manzoor, Wanling Cai, and Li Chen. 2021. A Survey on Conversational Recommender Systems. *Comput. Surveys* 54, 5 (2021), 1–36.

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

[22] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP '20)*. 6769–6781.

[23] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (2009), 30–37.

[24] Yehuda Koren, Steffen Rendle, and Robert Bell. 2022. Advances in Collaborative Filtering. *Recommender Systems Handbook* (2022), 91–142.

[25] Ivica Kostric, Krisztian Balog, and Filip Radlinski. 2021. Soliciting User Preferences in Conversational Recommender Systems via Usage-Related Questions. In *Fifteenth ACM Conference on Recommender Systems (RecSys '21)*. 724–729.

[26] Antonios Minas Krasakis, Andrew Yates, and Evangelos Kanoulas. 2022. Zero-Shot Query Contextualization for Conversational Search. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22)*. 1880–1884.

[27] Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226* (2018).

[28] Martin Josifoski Sebastian Riedel Luke Zettlemoyer Ledell Wu, Fabio Petroni. 2020. Zero-shot Entity Linking with Dense Entity Retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP '22)*. 6397–6407.

[29] Wenqiang Lei, Xiangnan He, Yisong Miao, Qingyun Wu, Richang Hong, Min-Yen Kan, and Tat-Seng Chua. 2020. Estimation-action-reflection: Towards deep interaction between conversational and recommender systems. In *Proceedings of the 13th International Conference on Web Search and Data Mining (WSDM '20)*. 304–312.

[30] Megan Leszczynski, Daniel Fu, Mayee Chen, and Christopher Re. 2022. TABi: Type-Aware Bi-Encoders for Open-Domain Entity Retrieval. In *Findings of the Association for Computational Linguistics: ACL 2022 (ACL Findings '22)*. 2147–2166.

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

[32] Sheng-Chieh Lin, Jheng-Hong Yang, Rodrigo Nogueira, Ming-Feng Tsai, Chuan-Ju Wang, and Jimmy Lin. 2020. Multi-Stage Conversational Passage Retrieval: An Approach to Fusing Term Importance Estimation and Neural Query Rewriting. arXiv:2005.02230 [cs.CL]

[33] Zeming Liu, Haifeng Wang, Zheng-Yu Niu, Hua Wu, and Wanxiang Che. 2021. DuRecDial 2.0: A Bilingual Parallel Corpus for Conversational Recommendation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP '21)*. 4335–4347.

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

[35] Kelong Mao, Zhicheng Dou, and Hongjin Qian. 2022. Curriculum Contrastive Context Denoising for Few-Shot Conversational Dense Retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22)*. 176–186.

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

[37] Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y. Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang. 2021. Large Dual Encoders Are Generalizable Retrievers. arXiv:2112.07899 [cs.IR]

[38] Javier Parapar and Filip Radlinski. 2021. Diverse User Preference Elicitation with Multi-Armed Bandits. In *Proceedings of the ACM international Conference on Web Search and Data Mining (WSDM '21)*. 130–138.

[39] Filip Radlinski, Krisztian Balog, Bill Byrne, and Karthik Krishnamoorthi. 2019. Coached Conversational Preference Elicitation: A case study in understanding movie preferences. In *SIGDial 2019*.

[40] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research* 21, 140 (2020), 1–67.

[41] Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive Learning Rates with Sublinear Memory Cost. arXiv:1804.04235 [cs.LG]

[42] Svitlana Vakulenko, Shayne Longpre, Zhucheng Tu, and Raviteja Anantha. 2021. Question Rewriting for Conversational Question Answering (WSDM '21). 355–363.

[43] Daniel Valcarce, Alejandro Bellogín, Javier Parapar, and Pablo Castells. 2018. On the Robustness and Discriminative Power of Information Retrieval Metrics for Top-N Recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems (RecSys '18)*. 260–268.

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

[45] Shi Yu, Jiahua Liu, Jingqin Yang, Chenyan Xiong, Paul Bennett, Jianfeng Gao, and Zhiyuan Liu. 2020. Few-Shot Generative Conversational Query Rewriting. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*. 1933–1936.

[46] Shi Yu, Zhenghao Liu, Chenyan Xiong, Tao Feng, and Zhiyuan Liu. 2021. Few-Shot Conversational Dense Retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*. 829–838.

[47] Hamed Zamani, Johanne R. Trippas, Jeff Dalton, and Filip Radlinski. 2022. Conversational Information Seeking. arXiv:2201.08808 [cs.IR]

[48] Shuo Zhang and Krisztian Balog. 2017. EntiTables: Smart Assistance for Entity-Focused Tables. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '17)*. 255–264.

[49] Xiangling Zhang, Yueguo Chen, Jun Chen, Xiaoyong Du, Ke Wang, and Ji-Rong Wen. 2017. Entity Set Expansion via Knowledge Graphs. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '17)*. 1101–1104.

[50] Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W Bruce Croft. 2018. Towards Conversational Search and Recommendation: System Ask, User Respond.

In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM '18)*. 177–186.

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

[52] Kun Zhou, Xiaolei Wang, Yuanhang Zhou, Chenzhan Shang, Yuan Cheng, Wayne Xin Zhao, Yaliang Li, and Ji-Rong Wen. 2021. CRSLab: An Open-Source Toolkit for Building Conversational Recommender System. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations (ACL-IJCNLP '21)*. 185–193.

[53] Kun Zhou, Yuanhang Zhou, Wayne Xin Zhao, Xiaoke Wang, and Ji-Rong Wen. 2020. Towards Topic-Guided Conversational Recommender System. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING '20)*. 4128–4139.

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

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