Retrieval-Augmented Conversational Recommendation with Prompt-based Semi-Structured Natural Language State Tracking

Sara Kemper*
University of Waterloo
Waterloo, Ontario, Canada

Justin Cui*
Kai Dicarlantonio*
Kathy Lin*
Danjie Tang*
University of Toronto
Toronto, Ontario, Canada

Anton Korikov
Scott Sanner
anton.korikov@mail.utoronto.ca
University of Toronto
Toronto, Ontario, Canada

ABSTRACT

Conversational recommendation (ConvRec) systems must understand rich and diverse natural language (NL) expressions of user preferences and intents, often communicated in an indirect manner (e.g., "*I'm watching my weight*"). Such complex utterances make retrieving relevant items challenging, especially if only using often incomplete or out-of-date metadata. Fortunately, many domains feature rich item reviews that cover standard metadata categories *and* offer complex opinions that might match a user's interests (e.g., "*classy joint for a date*"). However, only recently have large language models (LLMs) let us unlock the commonsense connections between user preference utterances and complex language in user-generated reviews. Further, LLMs enable novel paradigms for semi-structured dialogue state tracking, complex intent and preference understanding, *and* generating recommendations, explanations, and question answers. We thus introduce a novel technology *RA-Rec*, a **R**etrieval-**A**ugmented, LLM-driven dialogue state tracking system for Conv**Rec**, showcased with a video,[1] open source GitHub repository,[2] and interactive Google Colab notebook.[3]

CCS CONCEPTS

• **Information systems → Recommender systems**; **Personalization**; *Language models.*

KEYWORDS

Conversational Recommendation, LLM, Dialogue State Tracking

ACM Reference Format:
Sara Kemper, Justin Cui, Kai Dicarlantonio, Kathy Lin, Danjie Tang, Anton Korikov, and Scott Sanner. 2024. Retrieval-Augmented Conversational Recommendation with Prompt-based Semi-Structured Natural Language State Tracking. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24), July 14–18, 2024, Washington, DC, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3626772.3657670

*These authors contributed equally to this research.

SIGIR '24, July 14–18, 2024, Washington, DC, USA

ACM ISBN 979-8-4007-0431-4/24/07.
https://doi.org/10.1145/3626772.3657670

1 INTRODUCTION

Effective conversational recommendation (ConvRec) systems need to understand rich and diverse natural language (NL) expressions of user preferences and intents, often communicated in an indirect or subtle manner [12, 14, 17, 18]. For instance, a user who asks "*Do they have parking?*" is both *inquiring* and *providing a preference* for available parking. Similarly, a user looking for a restaurant who states "*I'm watching my weight*" is expressing a complex preference that requires commonsense reasoning and may not match any predefined restaurant metadata fields. Metadata is also often incomplete or out-of-date, making it challenging to connect NL requests to relevant item recommendations. This creates major limitations for traditional NL ConvRec systems that rely on mapping user intents and preferences to predefined metadata taxonomies [13, 19, 23, 26].

Fortunately, many recommendation domains have an abundance of rich NL item reviews that not only refer to standard metadata categories but also offer more complex opinions and narratives that might match a user's interests, e.g. *"The menu had lots of low-cal veggie options!"*. However, what we have lacked until recently with the advent of large language models (LLMs) [5, 6, 20] is the ability to unlock the commonsense reasoning connections between rich user preference utterances and expressive language in user-generated content such as NL reviews. In addition to bridging this language expression and reasoning gap, LLMs also provide novel opportunities to control and facilitate a range of interactions in ConvRec dialogue, such as understanding user intents and preferences, *and* generating recommendations, explanations, and answers to questions [8].

We thus introduce a novel open source demonstration technology *RA-Rec*, a **R**etrieval-**A**ugmented, LLM-driven dialogue state tracking system for Conv**Rec**, making the following contributions:

- We introduce prompt-driven ConvRec intent classification and state updating that captures nuanced NL expressions while maintaining domain-specific preference structure via a *semi-structured* NL dialogue state (Sec. 3.2).
- We extend recent work on *reviewed-item retrieval* [1] to ConvRec dialogue, generating state-based queries, recommendations, explanations, and question answers (Figure 2).
- We demonstrate *RA-Rec* for restaurant recommendation, including a video,[1] a well-documented open source GitHub repository under a *permissive* MIT License,[2] and an interactive Google Colab notebook that can run the system.[3]

[1]https://www.youtube.com/watch?v=W8Y56UW2LTU
[2]https://github.com/D3Mlab/llm-convrec
[3]https://apoj.short.gy/d3m-llm-convrec-demo

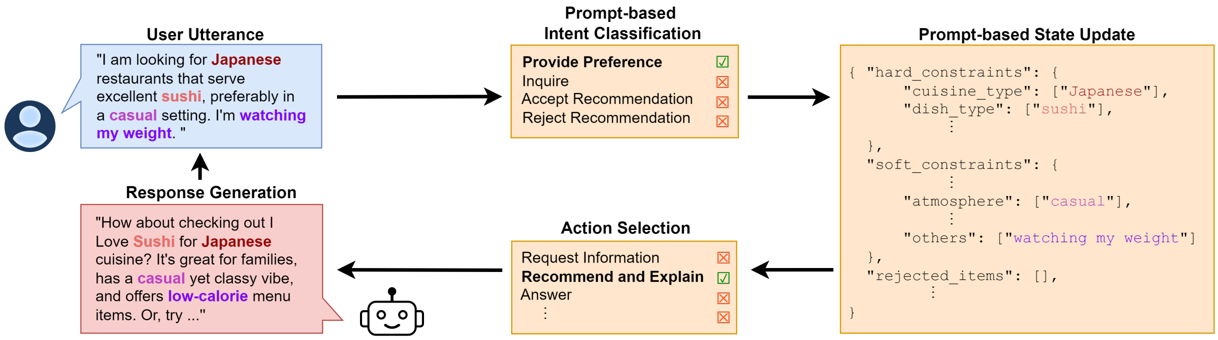

Figure 1: The *RA-Rec* prompt-driven dialogue state tracking loop. LLM prompting is used for multi-label intent classification and for updating a JSON semi-structured NL state which tracks user preferences and other key dialogue elements. The state keys provide an easily configurable structure, while LLM-generated state values can capture nuanced NL expressions.

Table 1: User intents and system actions in *RA-Rec*, which are a subset of the recommendation taxonomy of Lyu et al. [18].

User Intents		
Intent	**Description**	**Examples**
Provide Preference	Provide or refine preference for their desired item	"I want a place with a very good scenic view."
Inquire	Ask for more information about the recommended item(s)	"What kind of menu do they offer?", "How do these options compare for price?"
Reject Recommendation	Reject a recommended item, either explicitly or implicitly	"Probably too expensive, what else is there?"
Accept Recommendation	Accept a recommended item, either explicitly or implicitly	"The first place looks good!"
System Actions		
Action	**Description**	**Examples**
Request Information	Request the user's preferences towards item aspect(s)	"Where are you located?", "What kind of cuisine are you looking for?"
Recommend and Explain	Recommend item(s) and explain how they match user preferences	"How about trying Washoku Bistro for a comfortable and laid-back vibe while enjoying some delicious Japanese sushi?"
Answer	Respond to user inquiry about recommended item(s)	"Yes, Tokyo Express has a parking lot."
Respond to Rejection	Respond to user's rejection of recommended item(s)	"I'm sorry that you did not like the recommendation. Is there anything else I can assist you with?"
Respond to Acceptance	Respond to user's acceptance of recommended item(s)	"Great! If you need any more assistance, feel free to ask."
Greeting	Greet the user.	"Hello there! I am an Edmonton restaurant recommender. How can I help you?"

2 BACKGROUND AND RELATED WORK

2.1 Dialogue State Tracking

A standard Dialogue State Tracking (DST) loop [24] has four steps: (1) intent understanding, (2) dialogue state updating, (3) action selection, and (4) response generation. A traditional state consists of keys and values, typically from a predefined set of labels such as *"food: italian"*, *"price: cheap"*, *"area: east"*, that represent a most likely estimate of the participants' shared intentions and beliefs at a given turn [7, 24]. State tracking techniques generally map features extracted from user utterances to state labels, and include hand-crafted rules [3, 15], discriminative classifiers [4] and Bayesian networks [21, 25]. While following the DST loop steps for modular dialogue control, our *RA-Rec* system (Sec. 3) extends traditional DST methods with LLM-driven state tracking to capture complex, NL expressions of preference and to facilitate state-based retrieval-augmented recommendation and question answering (QA).

2.2 Reviewed Item Retrieval

Aiming to unlock the expressive NL data available in reviews, Abdollah Pour et al. [1] recently extended Neural IR [22] to an approach they call Reviewed Item Retrieval (RIR), where the key challenge lies in *fusing* low-level information from multiple reviews to a higher item level [28]. They demonstrate it is more effective to *first* score individual reviews against a query and *then* aggregate these scores to an item level (*late fusion*), instead of summarizing reviews at an item level before query-scoring (*early fusion*), since late fusion retains critical nuanced review information lost by early fusion.

In late fusion RIR, given a query and a set of reviews, a neural encoder maps each review and the query to respective embeddings. A similarity function, such as the dot product, then computes a query-review similarity score. For each item, scores from the reviews with the highest query-review similarities are then averaged (fused) to give a query-item similarity score, and the top-scoring items are returned. As we will discuss in the next section, our *RA-Rec* system adapts late fusion RIR to ConvRec by generating queries from a NL dialogue state and using review-based retrieval-augmented generation for recommendation and QA, as illustrated in Figure 2.

3 RETRIEVAL AUGMENTED CONVERSATIONAL RECOMMENDATION

To leverage both the modular structure of a traditional DST loop and the NL reasoning abilities of LLMs, we propose *RA-Rec*, a modular, retrieval-augmented ConvRec system, illustrated in Figure 1. We

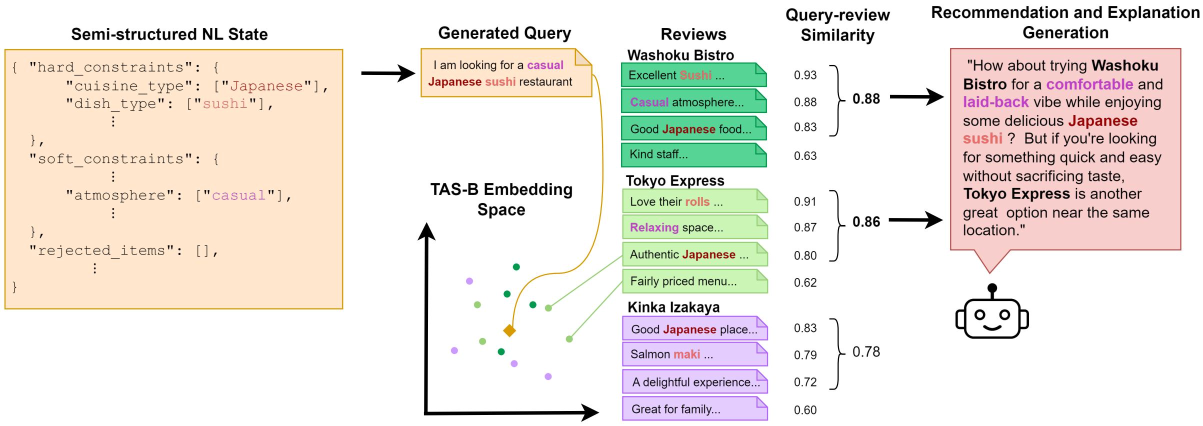

Figure 2: Retrieval-augmented recommendation and explanation using late fusion RIR. First, preferences in the dialogue state are used to generate a NL query. Then, query and review embeddings are scored using dot product similarity, and the top review scores for each item are averaged (fused) into an item score. Finally, the top items' most relevant reviews and metadata are used to generate a recommendation and explanation of how the item satisfies the preferences in the state.

Table 2: JSON keys in the *RA-Rec* semi-structured NL dialogue state. Subkeys can be modified to easily facilate new domains. Bold subkeys indicate mandatory preferences the system will request information about if these preferences are not provided.

State Key	Description	Subkeys
hard_constraints	User preferences that must be satisfied	**location**, **cuisine_type**, dish_type, price_range, atmosphere, dietary_restrictions, wait_times, type_of_meal, others
soft_constraints	User preferences that are not required	
recommended_items	Previously recommended items	-
rejected_items	Previously rejected items	-
accepted_items	Previously accepted items	-

employ a prompt-driven approach for intent classification and state updating, with the latter relying on a JSON format NL state that can be configured with domain-specific keys while capturing nuance through LLM-generated NL values. We then use this NL state to facilitate personalized, retrieval-augmented recommendation and QA utilizing item reviews and metadata.

3.1 Prompt-Driven Intent Classification

After the user makes an utterance, LLM-prompting is used to determine whether the user expresses any of the four intents in Table 1, which are a subset of the recommendation dialogue intent taxonomy of Lyu et al. [18]. Table 3 outlines the prompts used in *RA-Rec*, with full prompt templates available in the GitHub repository (see Sec. 1). We take a *multi-label* intent classification approach to capture multiple intents that might be expressed in a single utterance — for example, the utterance "*Does Washoku Bistro have parking?"* should be classified using both the intents *"Inquire"* and *"Provide preference"* because it expresses a preference towards available parking. A larger set of user intents can be facilitated by updating the system's prompts and initial state keys.

3.2 Semi-Structured NL Dialogue State Tracking

We store descriptions of user preferences and other important conversational elements such as rejected recommendations in a JSON state using the keys shown in Table 2 — two state examples are in Figures 1 and 2. While the *keys* provide structure, the state *values* are typically LLM-generated from the latest utterances, allowing the state to represent complex NL expressions of preference such as *"I'm watching my weight"* at a level of nuance and expressivity that would be impossible with predefined value sets. We thus refer to this state representation as a *semi-structured* NL dialogue state.

3.2.1 State Elements. Since the goal of *RA-Rec* is recommendation, the most important components of the state maintain an up-to-date belief about user preferences, represented through hard (required) and soft (not required) constraints. In our restaurant recommendation demo, these constraints are represented with several domain-specific subkeys listed in Table 2, as well as an "*others*" subkey to capture any unspecified preference types. To adapt *RA-Rec* to a new domain, these restaurant-specific subkeys can be replaced with domain-specific subkeys with little effort from a system designer.

Other state elements include previously recommended, rejected, or accepted restaurants – more elements could be easily added to handle a wider set of (domain-specific) user intents and system actions. Most state values are LLM-generated (prompts are summarized in Table 3) and used downstream for action selection, recommendation, explanation, and QA, discussed next.

3.3 Action Selection

The main system actions, summarized in Table 1 are *Request Information*, *Recommend and Explain*, and *Answer*. To understand our

Table 3: The main prompts used in *RA-Rec* – full templates can be found in the repository documentation (see Sec. 1 link).

Component	Prompt	Description
Intent Classification	Classify Intent	Given a user utterance and description of an intent (e.g. inquire), identify whether the utterance expresses the intent.
State Update	Update Constraints	Given a user utterance and the previous hard and soft constraints, update the constraints.
	Update Accepted/Rejected Item	Given a user utterance with intent "Accept/Reject Recommendation", identify which item was accepted/rejected.
Recommendation and Explanation	Generate Recommendation Query	Given hard/soft constraints in the state, generate a NL query.
	Explain Recommendations	Given the top retrieved items, their metadata, and their top reviews, explain how these recommended items match the hard/soft constraints.
QA	Determine QA Knowledge Source	Given a user inquiry about recommended items and those items' metadata, identify which fields should be used to answer the inquiry, if any. If none, reviews will be used as the QA knowledge source.
	Answer Using Metadata	Given an inquiry and relevant metadata entries, generate an answer.
	Generate QA Query	Given a user inquiry utterance, generate a NL query.
	Answer Using Reviews	Given an inquiry and retrieved reviews, generate an answer.

Request Information implementation, consider a user asking for a restaurant recommendation without giving a location preference — a recommendation may yield a restaurant in the wrong city! To avoid such premature recommendations with insufficient context, we identify *mandatory* preferences that the system must ask before recommending if not already provided by the user. In our demo, mandatory preferences are *location* and *cuisine_type* as shown in Table 2, but this selection is easily customized. Once mandatory preferences have been provided, the system will *Answer* if the user has made an inquiry and *Recommend and Explain* otherwise.

3.4 Retrieval-Augmented Recommendation *and* Explanation

To leverage expressive user review content in *RA-Rec*, we provide a novel adaptation of retrieval-augmented generation [16] for late fusion recommendation and explanation. To do this, we first generate a query based on semi-structured preferences in the dialogue state and then retrieve relevant items using RIR (Sec. 2.2) over both the item reviews and known metadata. This process is illustrated in Figure 2 with relevant prompts summarized in Table 3.

Specifically, after a NL query is generated from the hard and soft constraints in the state, we implement late fusion RIR to retrieve a list of top-k scoring items. Our implementation of late fusion RIR uses a TAS-B dense encoder [11] (a variant of BERT [9] fine-tuned for retrieval), dot product similarity, and approximate maximum-inner product search (MIPS) via FAISS [10] to enable scalability to large review corpora. After the top-k items ($k = 2$ in our demo) are retrieved, we use the metadata and top-scoring reviews for each item in a prompt to generate a recommendation and explanation of how these items match the dialogue state preferences.

3.5 Retrieval-Augmented Question Answering

As observed by Lyu et al. [18], the later stages of a recommendation conversational often involve a number of inquiries about the recommended item to confirm that it meets the user's requirements. To address such QA, *RA-Rec* retrieves relevant reviews or metadata for each of the items in question and uses this retrieved information to generate an answer – with Table 3 outlining the prompts used in our QA approach. Our framework is capable of addressing both *individual item questions* such as "*What kind of menu do they offer?*" as well as *comparative questions* such as "*How do their prices compare?*" as demonstrated in the video (see Sec. 1).

In more detail, the first step of QA uses prompting to determine whether an inquiry can be answered using available metadata, which is typically the best knowledge source for simple questions about common properties. In our restaurant recommendation demo, such common metadata fields include price, delivery availability, and parking information. If the inquiry cannot be answered with metadata, a NL query is generated from the user utterance and used to retrieve several reviews for each item in question. As discussed above, reviews are an expressive knowledge source, especially when inquiries and preferences are stated in complex ways. Finally, the retrieved reviews and metadata for each item are used to generate an answer to the question, which may include item comparisons.

3.6 *RA-Rec* System Summary

In summary, *RA-Rec* employs an LLM-driven, modular DST structure to facilitate a controllable recommendation dialogue that can connect complex NL user preferences to matching items using their reviews and metadata. Its JSON semi-structured NL state features configurable keys for domain-specific control while the LLM-updated state values are able to express NL nuance. This state supports novel retrieval-augmented recommendation, explanation, and QA, using scalable retrieval methods such as late fusion RIR and leveraging item reviews and metadata to generate responses.

4 DEMONSTRATION DETAILS

Our system is designed for easy adaptation to various domains, and as a demonstration, we present *RA-Rec* for restaurant recommendation — see Sec. 1 for demo links. Specifically, we use the Yelp Academic Dataset[4] to obtain metadata and over 46K reviews for 1298 restaurants in Edmonton, Alberta.[5] GPT-3.5-turbo is the LLM used for all prompting steps, but the *RA-Rec* framework is LLM-agnostic and will work with any prompt-based LLM model.

5 FUTURE WORK

RA-Rec is a flexible LLM prompt-driven architecture and thus opens many new directions for ConvRec systems to support natural user workflows [13, 18]. Key extensions include support for (1) active preference elicitation to narrow down large item spaces [2], (2) structured reasoning over multi-aspect NL preferences [27], and (3) trade-off negotiation between multiple recommendations.

[4]https://www.yelp.com/dataset/download
[5]The median number of reviews per restaurant was 21.

REFERENCES

[1] Mohammad Mahdi Abdollah Pour, Parsa Farinneya, Armin Toroghi, Anton Korikov, Ali Pesaranghader, Touqir Sajed, Manasa Bharadwaj, Borislav Mavrin, and Scott Sanner. 2023. Self-supervised Contrastive BERT Fine-tuning for Fusion-Based Reviewed-Item Retrieval. In *European Conference on Information Retrieval*. Springer, 3–17.

[2] David Eric Austin, Anton Korikov, Armin Toroghi, and Scott Sanner. 2024. Bayesian Optimization with LLM-Based Acquisition Functions for Natural Language Preference Elicitation. *arXiv preprint arXiv:xxxx.xxxxx* (2024).

[3] Dan Bohus and Alexander Rudnicky. 2003. Ravenclaw: Dialog management using hierarchical task decomposition and an expectation agenda. (2003).

[4] Dan Bohus and Alexander Rudnicky. 2006. A "k hypotheses+ other" belief updating model. (2006).

[5] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of Artificial General Intelligence: Early experiments with GPT-4. arXiv:2303.12712 [cs.CL]

[6] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. PALM: Scaling language modeling with pathways. 2022. *arXiv preprint arXiv:2204.02311* (2022).

[7] Philip R Cohen, Hector J Levesque, et al. 1987. *Rational interaction as the basis for communication.* CSLI Stanford.

[8] Yashar Deldjoo, Zhankui He, Julian McAuley, Anton Korikov, Scott Sanner, Arnau Ramisa, René Vidal, Maheswaran Sathiamoorthy, Atoosa Kasirzadeh, and Silvia Milano. 2024. A Review of Modern Recommender Systems Using Generative Models (Gen-RecSys). *arXiv preprint arXiv:2404.00579* (2024).

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. (Jun 2019), 4171–4186. https://doi.org/10.18653/v1/N19-1423

[10] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The FAISS library. (2024). arXiv:2401.08281 [cs.LG]

[11] Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 113–122.

[12] Mohammad Javad Hosseini, Filip Radlinski, Silvia Pareti, and Annie Louis. 2023. Resolving Indirect Referring Expressions for Entity Selection. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 12313–12335. https://doi.org/10.18653/v1/2023.acl-long.688

[13] Dietmar Jannach, Ahtsham Manzoor, Wanling Cai, and Li Chen. 2021. A survey on conversational recommender systems. *ACM Computing Surveys (CSUR)* 54, 5 (2021), 1–36.

[14] 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 *Proceedings of the Eleventh ACM Conference on Recommender Systems* (Como, Italy) *(RecSys '17)*. Association for Computing Machinery, New York, NY, USA, 229–237. https://doi.org/10.1145/3109859.3109873

[15] Staffan Larsson and David R Traum. 2000. Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Natural language engineering* 6, 3-4 (2000), 323–340.

[16] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (Vancouver, BC, Canada) *(NIPS'20)*. Curran Associates Inc., Red Hook, NY, USA, Article 793, 16 pages.

[17] Annie Louis, Dan Roth, and Filip Radlinski. 2020. "I'd rather just go to bed": Understanding Indirect Answers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, Online, 7411–7425. https://doi.org/10.18653/v1/2020.emnlp-main.601

[18] Shengnan Lyu, Arpit Rana, Scott Sanner, and Mohamed Reda Bouadjenek. 2021. A workflow analysis of context-driven conversational recommendation. In *Proceedings of the Web Conference 2021*. 866–877.

[19] Fedelucio Narducci, Pierpaolo Basile, Marco de Gemmis, Pasquale Lops, and Giovanni Semeraro. 2020. An investigation on the user interaction modes of conversational recommender systems for the music domain. *User Modeling and User-Adapted Interaction* 30 (2020), 251–284.

[20] TB OpenAI. 2022. ChatGPT: Optimizing language models for dialogue. *OpenAI* (2022).

[21] Tim Paek and Eric J Horvitz. 2013. Conversation as action under uncertainty. *arXiv preprint arXiv:1301.3883* (2013).

[22] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).

[23] Yueming Sun and Yi Zhang. 2018. Conversational recommender system. In *The 41st international acm sigir conference on research & development in information retrieval*. 235–244.

[24] Jason D Williams, Antoine Raux, and Matthew Henderson. 2016. The dialog state tracking challenge series: A review. *Dialogue & Discourse* 7, 3 (2016), 4–33.

[25] Jason D Williams and Steve Young. 2007. Partially observable Markov decision processes for spoken dialog systems. *Computer Speech & Language* 21, 2 (2007), 393–422.

[26] Zhao Yan, Nan Duan, Peng Chen, Ming Zhou, Jianshe Zhou, and Zhoujun Li. 2017. Building task-oriented dialogue systems for online shopping. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence* (San Francisco, California, USA) *(AAAI'17)*. AAAI Press, 4618–4625.

[27] Haochen Zhang, Anton Korikov, Parsa Farinneya, Mohammad Mahdi Abdollah Pour, Manasa Bharadwaj, Ali Pesaranghader, Xi Yu Huang, Yi Xin Lok, Zhaoqi Wang, Nathan Jones, et al. 2023. Recipe-MPR: A Test Collection for Evaluating Multi-aspect Preference-based Natural Language Retrieval. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2744–2753.

[28] Shuo Zhang and Krisztian Balog. 2017. Design patterns for fusion-based object retrieval. In *European Conference on Information Retrieval*. Springer, 684–690.