April 2020

Interactive Recommendation Engine for Conversational Recommendations

Anonymous

Interactive Recommendation Engine for Conversational Recommendations

ABSTRACT

The task of a traditional recommendation engine generally ends when it returns a list of possible answers to a user's query, even if the user finds the answers unhelpful. Such a paradigm is incompatible with modern virtual assistants, where a user might request to filter, control, and modify results in an interactive conversational flow that is based on user context. This disclosure describes a conversational recommendation engine that provides an explanation of the recommendations provided to the user, proactively offers the user options to find better recommendations, asks personalized follow-up questions to ease the interaction, and enables the user to provide real-time feedback to update the recommendation.

KEYWORDS

- Recommendation engine
- Interactive recommendation
- Conversational recommendation
- Conversational assistant
- Virtual assistant
- Smart speaker
- Smart appliance

BACKGROUND

A traditional recommendation engine responds to a user query by using a relatively complex algorithm to return a list of possible answers. As illustrated in the example below, the task of a traditional recommendation engine ends with the listing of possible answers, even when the user finds the answers unhelpful.

Example (traditional recommender)

User: "Show me some events."

Virtual assistant: "Here are some events." (The task of the recommender ends.)

However, such a paradigm is incompatible with modern virtual assistants, where a user might request to filter, control, and modify results in an interactive conversational flow.

DESCRIPTION

This disclosure describes a conversational recommendation engine that provides an explanation of its recommendations to the user, proactively offers the user options to find better recommendations, asks personalized follow-up questions to ease the interaction, and enables the user to provide real-time feedback to update the recommendation. In contrast to a traditional recommender, a conversational recommender functions in an interactive manner, as illustrated in the example below.

Example (conversational recommender)

User: "Show me some events."

Virtual assistant: "Here are some popular events close to you."

VA: "You went to several hiking events recently. Do you want to see more hiking events?"

U: "Yes."

VA: "Okay, check out these hiking events, the first one is also popular with your friends."

U: "Nice. Who is going to the first event?"

VA: "Your friends John and Mary are going."

U: "Great. I'll go to that event."

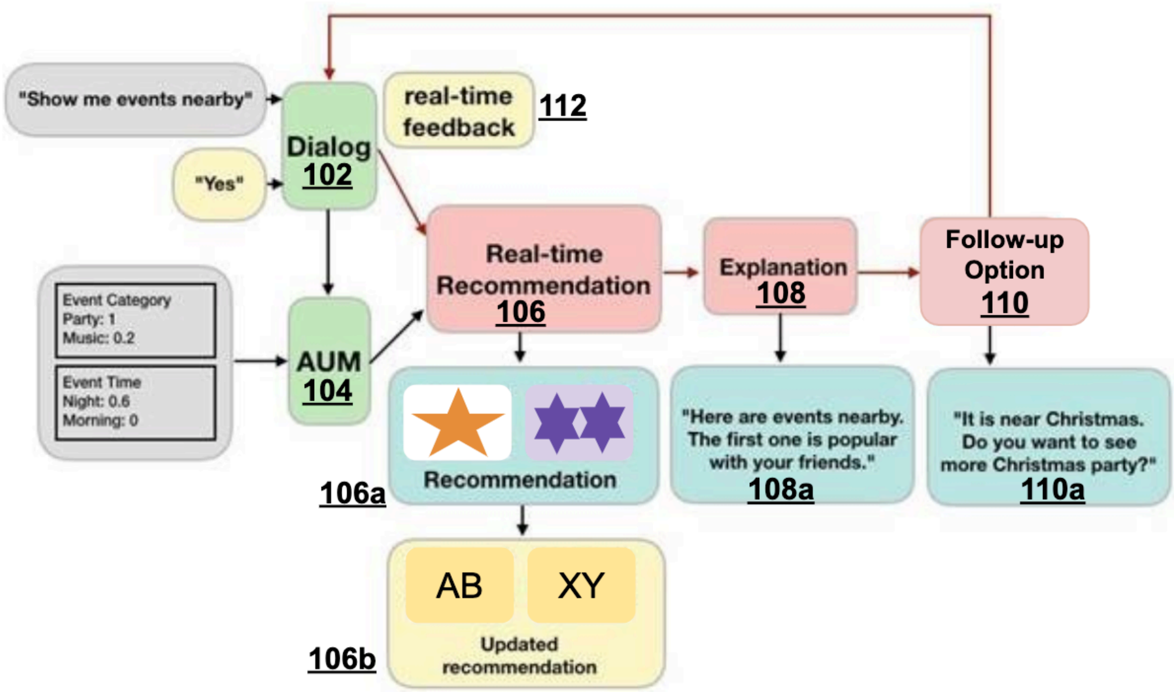

Fig. 1: Interactive recommendation engine

Fig. 1 illustrates an interactive recommendation engine, per the techniques of this disclosure. A dialog process or agent (102) handles conversation with the user. In this example, the user initiates a conversation, e.g., with a virtual assistant, with the query "show me events nearby." The virtual assistant user memory (AUM, 104) is a region of memory that stores user interest and context, e.g., events, their categories and other details, etc.

A real-time recommendation engine (106), described in greater detail below, draws from the dialog process and the AUM in order to issue an initial recommendation (106a). The real-time recommendation engine is capable of receiving and acting on real-time feedback (112), generated in the course of the conversation.

An explanation unit (108), described in greater detail below, provides a reason or context (108a) for the recommendations provided. In this example, the reason provided is “here are events nearby; the first one is popular with your friends.” A follow-up unit (110), described in greater detail below, provides follow-up options (110a), e.g., “it is near Christmas; do you want to see more Christmas parties?” The explanations and follow-up options inserted into the conversation may generate real-time feedback (112) from the user. Such feedback is provided to the recommendation engine which can use it to issue an updated recommendation (106b).

The components of the interactive recommendation model are explained in greater detail below.

Follow-up unit

The follow-up unit can generate options in various forms, e.g., a question or a tip that can be shown by the dialog agent. Follow-up options are a proactive way to get feedback from the user. Follow-up options can perform one or more of the following functions:

- *Teach the user about the requirements that they can add in a query to the assistant*: For example, when the user only knows to ask “show me some events”, the follow-up option can teach the user to add requirements by asking follow-up questions such as “which category of events do you like?”, “would you like to see events popular with your friends?”, or showing them follow-up tips such as “try ‘show me hiking events.’”
- *Offer the user personalized suggestions to improve the recommendation*: If the recommender finds a filter condition that matches the user’s interest and can improve the recommendation results, it can ask the user a follow-up question to confirm the filter condition. For example, “it’s near Christmas, do you want to see more Christmas events?”, “you went to several hiking events recently, do you want to see more hiking

events?" Such personalized suggestions help the user find more events that they are interested in.

- *Offer the user a personalized relaxation option when there is no match*: A user that is experienced in interacting with the virtual assistant knows to create concrete filter options, e.g., "show me hiking events this weekend, popular with my friends." Rather than provide an uninformative "sorry, there is no event satisfying your requirement," follow-up questions can provide personalized relaxation options, e.g., "sorry, there are no events popular with your friends; would you like to see some hiking events this weekend instead?" Providing such follow-up options reduces or eliminates the time needed by the user to modify the initial query.

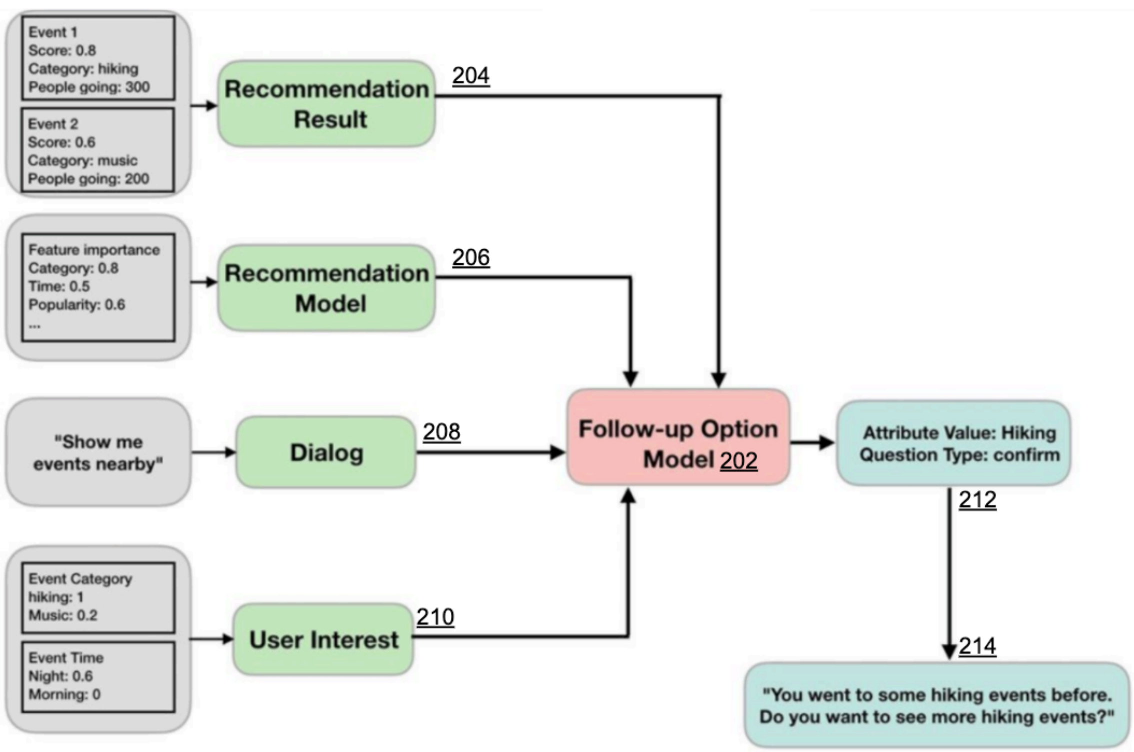

Fig. 2: A model for the follow-up option unit

Fig. 2 illustrates an example model for the follow-up option unit (202). The follow-up unit uses the current state of the recommendation engine, e.g., recent recommendation results (204); features input to the recommendation model (206); the on-going dialog (208) between user and virtual assistant; the user's interest (210); etc., to determine attributes (212) of the follow-up options before selecting and providing follow-up options (214). The follow-up options offered to the user are efficient and personalized, generated by taking into account user interests, requests, and context. The user's interests and previous interactions can be retrieved from the AUM. In this manner, the follow-up unit optimizes the follow-up options provided to the user to continue the conversation such that the user can determine a useful recommendation in a relatively short time.

Real-time recommendation engine

With follow-up options provided as described above, users can provide feedback to the recommender to improve the recommendation. The interactive recommendation engine updates recommendation results based on the dialog feedback from the user. Results are based on the user's experience in a multi-turn recommendation rather than a single-turn recommendation. In this manner, recommendations can be issued for requests (e.g., "show me hiking events") on topics for which the user has not shown recent or historic interest.

Explanation unit

While the follow-up unit proactively generates user feedback, the explanation unit provides the user with reasons for the recommendations. In consort with the follow-up unit, the explanation unit elicits a greater volume and quality of user feedback. The user feedback, as reflected within the dialog process, is used to adjust recommendation results and provide newer and more relevant recommendations. In this manner, the explanation unit provides for a better user experience. Some examples follow; the italicized text represents explanations for the recommendations provided.

Example 1

User: “Show me some events.”

Virtual assistant: “Here are some events, *popular and close to you*.”

Example 2

U: “Show me some music events.”

VA: “There is a *popular* musical *which is happening this weekend*.”

Example 3

U: “Show me some events.”

VA: “I found some events you might be interested *based on your previous interactions with me*.”

In this manner, the techniques of this disclosure enable users to explicitly control the results and understand the reasons for the recommendations issued by a recommendation engine. The follow-up options described herein offers users an easy to use interface to improve the recommendations received and teaches users to effectively query the virtual assistant.

CONCLUSION

This disclosure describes a conversational recommendation engine that provides an explanation of the recommendations provided to the user, proactively offers the user options to find better recommendations, asks personalized follow-up questions to ease the interaction, and enables the user to provide real-time feedback to update the recommendation.

REFERENCES

[1] Thompson, Cynthia A., Mehmet H. Goker, and Pat Langley. "A personalized system for conversational recommendations." https://arxiv.org/pdf/1107.0029.pdf accessed on Mar. 30, 2020.

[2] Yao Ming, Panpan Xu, Huamin Qu, and Liu Ren. "Interpretable and steerable sequence learning via prototypes. "In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 903-913. 2019. https://lliquid.github.io/homepage/files/kdd19_sequence_prototypes.pdf accessed on Mar. 30, 2020.

[3] Suengwhan Moon, Parath Shah, Anuj Kumar, and Rajen Subba, "OpenDialKG: Explainable conversational reasoning with attention-based walks over knowledge graphs" In *Proc. 57th Ann. Meeting Assoc. Computational Linguistics,* pp 845-54, Florence Italy, Jul. 28-Aug. 2, 2019 https://www.aclweb.org/anthology/P19-1081.pdf accessed on Mar. 30, 2020.

[4] Jianbo Chen, Le Song, Martin J. Wainwright, and Michael I. Jordan, "Learning to explain: An information-theoretic perspective on model interpretation." https://arxiv.org/pdf/1802.07814.pdf accessed on Mar. 30, 2020.

[5] Christakopoulou, Konstantina, Alex Beutel, Rui Li, Sagar Jain, and Ed H. Chi. "Q&R: A two-stage approach toward interactive recommendation." In *Proc. the 24th ACM SIGKDD Intl. Conf. on Knowledge Discovery & Data Mining*, pp. 139-148. 2018. http://www.alexbeutel.com/papers/q-and-r-kdd2018.pdf accessed on Mar. 30, 2020.

[6] Wang, Hongwei, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. “RippleNet: propagating user preferences on the knowledge graph for recommender systems” https://arxiv.org/pdf/1803.03467.pdf accessed on Mar. 30, 2020.

[7] Zhang, Yongfeng, and Xu Chen. “Explainable recommendation: A survey and new perspectives.” https://arxiv.org/pdf/1804.11192.pdf accessed on Mar. 30, 2020.

[8] Zhang, Xiaoying, Hong Xie, Hang Li, and John Lui. “Toward building conversational recommender systems: A contextual bandit approach.” https://arxiv.org/pdf/1906.01219.pdf accessed on Mar. 30, 2020.

[9] Sun, Yueming, and Yi Zhang. “Conversational recommender system.” https://arxiv.org/pdf/1806.03277.pdf accessed on Mar. 30, 2020.

[10] Li, Raymond, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. "Towards deep conversational recommendations." In *Advances in neural information processing systems*, pp. 9725-9735. 2018. http://papers.nips.cc/paper/8180-towards-deep-conversational-recommendations.pdf accessed on Mar. 30, 2020.