Improving Conversational Recommender Systems via Transformer-based Sequential Modelling

Jie Zou
SCALE@NTU & Nanyang Technological University
Singapore
jie.zou@ntu.edu.sg

Evangelos Kanoulas
University of Amsterdam
Amsterdam, The Netherlands
e.kanoulas@uva.nl

Pengjie Ren
Shandong University
Qingdao, China
renpengjie@sdu.edu.cn

Zhaochun Ren
Shandong University
Qingdao, China
zhaochun.ren@sdu.edu.cn

Aixin Sun
SCALE@NTU & Nanyang Technological University
Singapore
AXSun@ntu.edu.sg

Cheng Long
SCALE@NTU & Nanyang Technological University
Singapore
c.long@ntu.edu.sg

ABSTRACT

In Conversational Recommender Systems (CRSs), conversations usually involve a set of related items and entities e.g., attributes of items. These items and entities are mentioned in order following the development of a dialogue. In other words, potential sequential dependencies exist in conversations. However, most of the existing CRSs neglect these potential sequential dependencies. In this paper, we propose a Transformer-based sequential conversational recommendation method, named TSCR, which models the sequential dependencies in the conversations to improve CRS. We represent conversations by items and entities, and construct user sequences to discover user preferences by considering both mentioned items and entities. Based on the constructed sequences, we deploy a Cloze task to predict the recommended items along a sequence. Experimental results demonstrate that our TSCR model significantly outperforms state-of-the-art baselines.

CCS CONCEPTS

• **Information systems → Recommender systems**; Question answering.

KEYWORDS

Conversational recommendation; Sequential recommendation; Transformer

ACM Reference Format:
Jie Zou, Evangelos Kanoulas, Pengjie Ren, Zhaochun Ren, Aixin Sun, and Cheng Long. 2022. Improving Conversational Recommender Systems via Transformer-based Sequential Modelling. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3477495.3531852

SIGIR '22, July 11–15, 2022, Madrid, Spain

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3531852

Role	Message
Seeker	Hi I am looking for a good thriller really, any type new or old.
Recommender	Hi there! Do you want action or suspense?
Seeker	Yes I do.
Recommender	My favorite thrillers are the sequels to Fast & Furious 1.
Seeker	Yeah that one is good.
Recommender	The best one is the fourth one. It's called Fast & Furious 4, it's full of action with an addticting plot.
Seeker	I love those. Have you seen Fast & Furious 8 yet?
…	…

Figure 1: An example dialogue from the ReDial dataset. The mentioned items (i.e., movies) are highlighted in blue color, and entities in red color.

1 INTRODUCTION

In general, a recommender system learns user preference from historical user-item interactions, and then recommends items of user's preference. The recommended items can be delivered to users through various interfaces depending on the task, e.g., list of recommended products on e-commerce websites. Thanks to the rapid development of chatbots, CRS is now becoming a promising interface to deliver recommended items to users directly through dialogues.

An example dialogue is shown in Fig. 1, and our task is to recommend movies (i.e., the items) to users. Observe that there are two important properties demonstrated in this dialogue. First is that it is common to mention entities that are closely related to the items to be recommended, e.g., director and genre of a movie in our example. Second, the mentioned items (and also entities) naturally form a sequence, following the development of the conversation. That is, there is an order impact among items in a conversation and potential sequential dependencies within the conversation. For instance, when people talk about a movie (e.g., Fast & Furious 1), it is natural and reasonable to recommend a sequel or prequel of that movie (e.g., Fast & Furious 4). We argue that the modeling of such sequential dependency can well capture the context of users' activities, and has great potential to improve the quality of recommendations.

Generally, a CRS integrates two modules: a recommender module, and a dialogue module. The dialogue module generates natural language conversations to interact with users. The recommender module focuses on recommending desirable items to users by utilizing the information from the conversation, as well as related information from external sources like knowledge bases. In this work, we focus on the recommender module only.

As a rapidly growing research topic in recent years, a number of solutions for CRS have been proposed [5, 12, 13, 32]. One attempt is based on the "system asks – user responds" mode and simulates conversations by using some "anchor" text, e.g., item aspects [26], entities [31–35], facets or attributes [7, 8, 17, 29]. They usually utilize a belief tracker to infer the anchor-based preferences to improve CRS. Another mainstream approach is based on human-generated dialogues [1, 2, 10, 11, 14, 20, 21, 24, 25, 28, 30]. For instance, at the early stage, Li et al. [9] proposed a CRS dataset, ReDial, and presented a benchmark model for item recommendation. ReDial soon became the most widely used dataset for CRS. Given that entities within ReDial utterances are linked to a knowledge base, most subsequent work uses knowledge graphs to improve CRS [2, 11, 14, 27, 28, 30]. Although the aforementioned work demonstrates success to some extent, they neglect the *order impact* among items or entities discussed earlier. Hence, these existing solutions do not model the potential sequential dependency within the conversations.

Inspired by the success of Transformer methods like BERT [3], we explicitly model the bidirectional sequential dependency in conversations by using Transformer. Recent studies have shown that a carefully designed task-specific input format to BERT could lead to state-of-the-art performance [22]. Our solution is along this line. In this paper, we propose a simple and effective Transformer-based Sequential Conversational Recommender (TSCR) model for CRSs. Specifically, we extract both the mentioned items and entities in conversations as contextual information to construct a user sequence. Based on this user sequence, we randomly mask some items and deploy a Cloze task [3, 18] to predict the masked items by leveraging the bidirectional contextual information in the input sequence. The bidirectional representation for the user sequence is modeled by the deep bidirectional self-attention architecture. Our experiments on the conversational recommendation dataset demonstrate that our TSCR model significantly outperforms various state-of-the-art baselines.

2 SEQUENTIAL CONVERSATIONAL RECOMMENDER SYSTEM

We suppose there is a set of users $\mathcal{U} = \{u_1, u_2, \ldots, u_{|\mathcal{U}|}\}$, items $\mathcal{V} = \{v_1, v_2, \ldots, v_{|\mathcal{V}|}\}$, and conversations $\mathcal{D}$. We extract entities $\mathcal{E} = \{e_1, e_2, \ldots, e_{|\mathcal{E}|}\}$ from conversations $\mathcal{D}$ based on DBpedia. For a user $u \in \mathcal{U}$, we have his/her item/entity mention history, extracted from his/her conversation, denoted as a sequence $\mathcal{S}_u = [s_1^u, \ldots, s_k^u, \ldots, s_K^u]$ ($s_k^u \in \{\mathcal{V} \cup \mathcal{E}\}$), we aim to accurately predict the next item v^* that user u likes, along the development of the conversation. In the following, we first describe the base model, i.e., Transformer [16, 19], adopted in CRS, and then describe how we train our model and perform the item recommendation.

2.1 Base Model

Inspired by Sun et al. [16], we adopt Transformer [16, 19] as our base model, which consists of the embedding layer, self-attention layer, and prediction layer.

Embedding layer. Given a sequence, we denote the embedding for the element at position k in the input sequence as $\mathbf{h}_k^0$. For the representation of $\mathbf{h}_k^0$, we inject a learnable position embedding, $\mathbf{p}_k$, into the embedding of each element of the input sequence, $\mathbf{s}_k$

$$\mathbf{h}_k^0 = \mathbf{s}_k + \mathbf{p}_k. \tag{1}$$

All elements together form a trainable embedding matrix $\mathbf{H}^0$. Based on this initially trainable embedding matrix $\mathbf{H}^0$, we interactively calculate $\mathbf{H}^n$ at each Transformer layer n.

Self-attention layer. A self-attention layer consists of two sublayers: a multi-head self-attention sub-layer and a Position-wise Feed-Forward Network (PFFN). More details can be found in Vaswani et al. [19].

$$\mathbf{H}^{n+1} = \mathrm{MultiHead}(\mathrm{PFFN}(\mathbf{H}^n)). \tag{2}$$

We construct the PFFN by the Feed-Forward Network (FFN) with GELU activation [4] at each position separately:

$$\mathrm{PFFN}(\mathbf{H}^n) = [\mathrm{FFN}(\mathbf{h}_1^n)^\top; \ldots; \mathrm{FFN}(\mathbf{h}_k^n)^\top]^\top. \tag{3}$$

In addition, we deploy a residual connection around each of the two sub-layers, followed by a dropout and layer normalization, i.e., the output of each sub-layer is actually:

$$\mathrm{LayerNorm}(\mathbf{H}^n + \mathrm{Dropout}(\mathrm{sublayer}(\mathbf{H}^n))), \tag{4}$$

where sub-layer is MultiHead or PFFN in Eq. 2.

Prediction layer. After N layers of Transformer, we get the final output $\mathbf{H}^N$ for the input sequence. Assuming we mask s_k at the input sequence, we then utilize $\mathbf{h}_k^N$ to predict the masked item s_k. Specifically, we apply a softmax function through a two-layer feed-forward network with GELU activation in between to produce an output distribution over items. To ensure recommendations are all items, we set the score of non-item entities in the softmax function to $-\infty$.

2.2 Masked Item Prediction

We apply a Cloze task [3, 18] on the item/entity mention sequence from the conversational history of a user to train our model. Given a sequence $\mathcal{S}_u$, we randomly mask a proportion of items (we only mask items and never entities given that our target is predicting items) in the input sequence, by replacing them with the special token "[mask]", and then predict the original IDs of the masked items (In our implementation, items are represented by their corresponding item IDs.). Following BERT [3], we leverage the bidirectional contextual information in the input sequence for predicting the masked item. We use the negative log-likelihood of the masked targets as the loss:

$$\mathcal{L} = \frac{1}{|\mathcal{S}_u^{(mask)}|} \sum_{v' \in \mathcal{S}_u^{(mask)}} -\log P(v' | \mathcal{S}_u'), \tag{5}$$

where $\mathcal{S}_u'$ is the masked version for user historical sequence $\mathcal{S}_u$, $\mathcal{S}_u^{(mask)}$ is the set of masked items in $\mathcal{S}_u$, and v' is one of the masked items.

For testing, it is not practical to use the bidirectional information to predict as the testing item is always in the future given the current context. To this end, we construct a contextual sequence for each testing item, and then add a "[mask]" token to the end of the sequence to predict a testing item. For example, if there are three items in a sequence, we mask the first item and predict it with possible entities that are already mentioned in the dialogue till this prediction. Then we mask and predict the second item based on the first item and entities mentioned so far in the dialogue till this prediction, then the third item. To better match the last item prediction during testing, we also mask the last item for each training sequence to generate a training sample during training.

Contrary to bidirectional Transformer models like BERT [19], which is a pre-training model for sentence representation, our TSCR model is an end-to-end model trained for sequential conversational recommendation. Also, we removed the next sentence loss and next sentence prediction since there is only one sequence of user's historical item/entity mentions in CRS. Different from those works using bidirectional Transformer or pre-trained BERT for recommender systems, we trained our TSCR model in an end-to-end style and incorporated the conversational information (e.g., entities), aiming to improve CRS.

3 EXPERIMENTS AND ANALYSIS

Through experiments, we aim to answer the following research questions:

RQ1 How effective is our proposed simple model compared to current state-of-the-art?
RQ2 What are the contributions of items and entities in a sequence?
RQ3 What is the effect of item position?
RQ4 How do the parameters of our proposed model affect its efficacy?

3.1 Experimental Setting

Dataset. In this work, we use the conversational recommendation dataset *REcommendations through DIALog (ReDial)* to evaluate our model, same as Chen et al. [2], Li et al. [9], Sarkar et al. [14], Zhou et al. [28], and Ma et al. [11]. ReDial is a set of annotated dialogues in which a seeker requests movie suggestions from the recommender. It contains 956 users, 51,699 movies, 10,006 conversations, and 182,150 utterances. The dataset is split into training, validation, and test sets by 8:1:1 ratio. Besides movies (i.e., the items), we extract the relevant entities, such as director and genre, from DBpedia, as suggested by Chen et al. [2], Sarkar et al. [14], Zhou et al. [28], and Ma et al. [11].

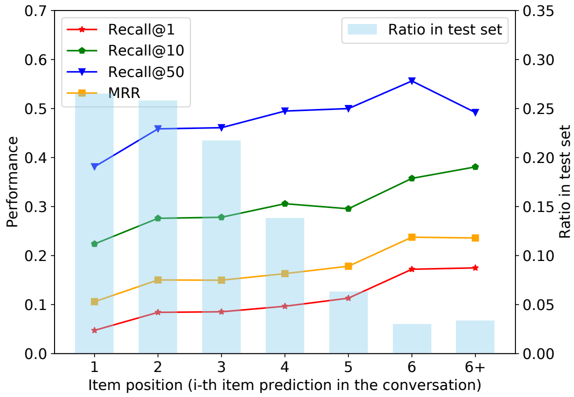
Evaluation Metrics. Similar to Chen et al. [2], Zhou et al. [28], and Ma et al. [11], we use Recall@k (k = 1, 10, and 50) as our evaluation metrics for the recommendation task in CRSs. Recall@k evaluate whether the target item provided by human recommenders appears in the top-k items produced by the recommender system. Moreover, we use the Mean Reciprocal Rank (MRR) to indicate the mean of the reciprocal of the rank of the target item in the ranked list predicted by the model. For each conversation, we start from the first item (movie) to recommend in the recommender's responses. This means, each item in the recommender's responses is regarded as ground truth and we evaluate them one by one throughout the conversation following the previous work [2, 11, 28]. For each testing instance, we rank all possible items within the dataset.

Table 1: Recommendation performances between our model TSCR and baselines. '*' indicates significant improvements upon the best baseline in Fisher random test with p-value < 0.05. Best performances are in bold.

Model	Recall@1	Recall@10	Recall@50
Popularity	0.012	0.061	0.179
TextCNN	0.013	0.068	0.191
REDIAL	0.023	0.129	0.287
KBRD	0.030	0.164	0.338
COLING20	0.034	0.181	0.357
KGSF	0.039	0.183	0.378
CR-Walker	0.040	0.187	0.376
CRFR	0.040	0.202	0.399
TSCR	**0.075***	**0.262***	**0.444***
-w/o entity	0.072	0.255	0.436
-w/o item	0.033	0.148	0.320

Parameter Settings. We train our model using Adam [6] and TensorFlow with a learning rate of 1e-4. We set the batch size = 256, layer number N = 2, head number = 2, the maximum sequence length K = 50, L2 regularization strength = 0.01, and the global norm clip of gradients = 5 for stable training. We study the effect of the hidden dimensionality and mask proportion in Section 3.5. The hidden dimensionality ranges within [32, 64, 128, 256] and the mask proportion is tuned within the range of [0.2, 0.4, 0.6, 0.8]. For the parameter settings of all baselines, we use the results of each baseline under its optimal hyper-parameter settings.

Baseline. In this work, we consider two classical baselines and several strong baselines used against ReDial: (1) **Popularity** is a classical baseline sorting the items according to historical recommendation frequency. (2) **TextCNN** is a classical CNN-based recommendation model learning embeddings from contextual utterances. (3) **ReDial** [9] is the benchmark model of ReDial applying an autoencoder recommender for the conversational recommendation. (4) **KBRD** [2] utilizes the DBpedia knowledge graph to introduce knowledge-grounded information to improve conversational recommendation. (5) **COLING20** [14] is based on KBRD and constructs subgraphs of the DBpedia knowledge graph to improve recommendation performance for the conversational recommendation. (6) **KGSF** [28] incorporates a knowledge graph enhanced recommender by utilizing both entity-oriented and word-oriented knowledge graphs. (7) **CR-Walker** [11] is one of the state-of-the-art conversational recommendation models by performing tree-structured reasoning on the knowledge graph. (8) **CRFR** [27] is one of the state-of-the-art conversational recommendation models based on reinforcement learning and multi-hop reasoning on knowledge graphs.



Item position	1	2	3	4	5	6	6+
Sequence length	1.3	3.1	4.9	6.2	7.9	9.6	13.3

Figure 2: The performance of TSCR with the ordinal number of item predictions.

3.2 Overall Performance (RQ1)

In this section, we study how effective is our proposed method compared to prior solutions. We compare our recommendation performances with baselines as shown in Table 1. The evaluation metrics are reported as the average performance for the max number of conversational turns. From Table 1, we observe that REDIAL outperforms the classical recommendation models, Popularity and TextCNN, by using mentioned items in the dialogue to make recommendations. Furthermore, KBRD, COLING20, KGSF, CR-Walker and CRFR outperform REDIAL, which might be because they introduce external knowledge graphs and entities to understand the user's intentions. Also, we see that our proposed model, TSCR, significantly outperforms all the baselines on all three metrics. Take Recall@50 as an example, we outperform ReDial, KBRD, COLING20, KGSF, CR-Walker and CRFR by 55%, 31%, 25%, 18%, 18%, and 11%, respectively. This indicates that our TSCR model is effective and incorporating the sequential occurrence of items and entities is highly beneficial for improving recommender performance in CRS. But note that this does not mean KBRD, KGSF, CR-Walker and CRFR are worse than our TSCR model, as they focus more on the natural language response generation part and need to balance the recommender part and natural language response generation part by jointly modeling them in CRS. In addition, different from most prior solutions which use knowledge graphs to reduce candidate item space [14], we do not make use of the structure of knowledge graphs, although a sequence in this work can be mapped to a path in knowledge graphs. In other words, we only use the knowledge base as a dictionary to extract the relevant entities for the items mentioned in a dialogue.

3.3 Impact of Entity and Item in Sequence (RQ2)

To understand what are the contributions of different model components, we conduct an ablation study comparing our model with its ablation variants (TSCR removing the entity mentions "-w/o entity" and TSCR removing the item mentions "-w/o item"). The results are shown in Table 1. It can be seen from Table 1 that both the item mentions and entity mentions in the conversation contribute to the final performance. After removing item mentions or entity mentions from the context, the recommendation performance on all three metrics drops, which indicates the importance of the two components. Also, we observe that item mentions contribute more than entity mentions. This might be because that item mentions are more reflective of user true preferences and entity mentions contain more noise than item mentions. This suggests that sentiment analysis for entity mentions to distill the sequence of entity mentions might be beneficial [23].

3.4 Effect of Item Position (RQ3)

In this section, we explore whether the item position in the conversation affects the recommender performance. We compare the recommender performance for each position of item predictions, from 1-st item prediction to 6+ item prediction in the conversation. From Fig. 2, we observe that, most of dialogues (74.1%) contain only 1–3 item recommendations. This is in line with that users expect the system can perform high-quality recommendations with fewer rounds in real applications. Overall, Fig. 2 shows that the performance of our TSCR model improves as the item position increases. We attribute this to the fact that the model collects more context information about the user as the item position increases. Higher item position means longer session sequence length of item and entity mentions. This indicates that the TSCR performance improves when the sequence length gets longer. Specifically, when the recommender suggests the first item (i.e., the item position is equal to 1, corresponding to the classical problem "cold start" [15]), the TSCR recommender can still achieve high performance based on the contextual information.

3.5 Parameter Sensitivity (RQ4)

Effect of hidden dimensionality. We now explore how the hidden dimensionality affects the model performance. As shown in Fig. 3 (left), we observe the recommendation performance of our TSCR model decreases with the embedding size increases. This is probably because of over-fitting. The TSCR model achieves the best performance when the hidden dimensionality is equal to 32.

Effect of mask proportion. Fig. 3 (right) shows how the mask proportion affects the model performance. It can be seen that from Fig. 3, our TSCR model performs stably with the change of the mask proportion. The best performing mask proportions are 0.4–0.6.

4 CONCLUSION AND FUTURE WORK

In this work, we have proposed the Transformer-based Sequential Conversational Recommender (TSCR) for CRSs. TSCR deploys a Cloze task and models the sequential dependency of both items and entities in conversations by the deep bidirectional self-attention architecture. Our model uses the knowledge base as a dictionary to get related entities, but does not use the structure of the knowledge base for any reasoning, making it simple and straightforward. Experimental results on the ReDial conversational recommendation dataset show that our TSCR model, despite simple is highly effective, constituting a very strong baseline for future researchers to use.

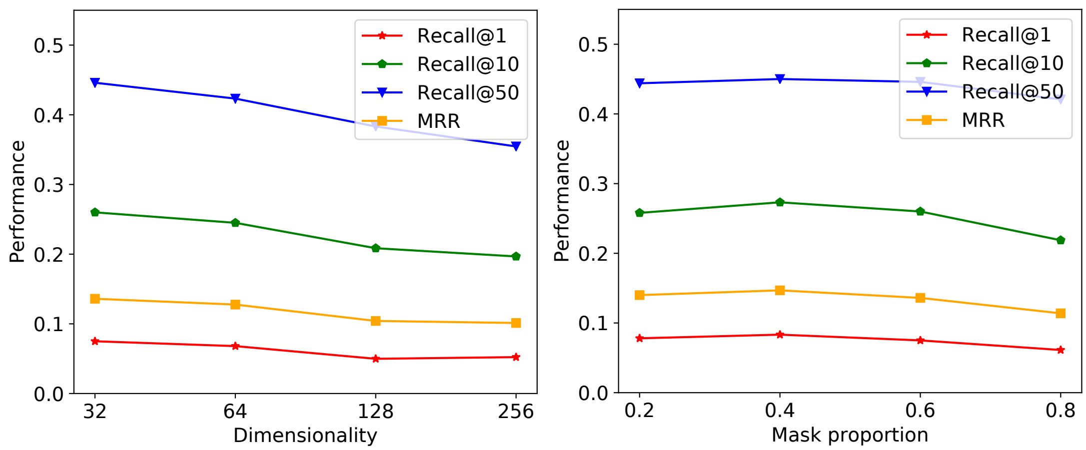


Figure 3: Effect of hidden dimensionality (left) and mask proportion (right).

One limitation of this work is that we only focus on the recommender module. As for future work, we plan to incorporate the natural language response generation part as well. Moreover, in this work we do not model the sentiment of mentioned items or entities and treat them as the same in the conversation. It is worth exploring the CRS by incorporating and modeling the sentiment (e.g., positive or negative) from users' feedback in future work.

5 ACKNOWLEDGEMENT

This study is supported under the RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from Singapore Telecommunications Limited (Singtel), through Singtel Cognitive and Artificial Intelligence Lab for Enterprises (SCALE@NTU). This work is also supported by the Natural Science Foundation of China (61902219, 62072279, 62102234), the Natural Science Foundation of Shandong Province (ZR2021QF129), Shandong University multidisciplinary research and innovation team of young scholars (No. 2020QNQT017), Meituan, the Tencent WeChat Rhino-Bird Focused Research Program (JR-WXG-2021411), the NWO Smart Culture - Big Data / Digital Humanities (314-99-301), the NWO Innovational Research Incentives Scheme Vidi (016.Vidi.189.039), and the H2020-EU.3.4. - SOCIETAL CHALLENGES - Smart, Green And Integrated Transport (814961). All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

REFERENCES

[1] Keyu Chen and Shiliang Sun. 2021. Knowledge-Based Conversational Recommender Systems Enhanced by Dialogue Policy Learning. In *The 10th International Joint Conference on Knowledge Graphs*. 10–18.

[2] Qibin Chen, Junyang Lin, Yichang Zhang, Ming Ding, Yukuo Cen, Hongxia Yang, and Jie Tang. 2019. Towards Knowledge-Based Recommender Dialog System. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 1803–1813.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[4] Dan Hendrycks and Kevin Gimpel. 2016. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. (2016).

[5] Dietmar Jannach and Ahtsham Manzoor. 2020. End-to-End Learning for Conversational Recommendation: A Long Way to Go?. In *IntRS@ RecSys*. 72–76.

[6] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[7] 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*. 304–312.

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

[9] Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards Deep Conversational Recommendations. In *Advances in Neural Information Processing Systems*. 9748–9758.

[10] Yu Lu, Junwei Bao, Yan Song, Zichen Ma, Shuguang Cui, Youzheng Wu, and Xiaodong He. 2021. RevCore: Review-augmented conversational recommendation. *arXiv preprint arXiv:2106.00957* (2021).

[11] Wenchang Ma, Ryuichi Takanobu, and Minlie Huang. 2021. CR-Walker: Tree-Structured Graph Reasoning and Dialog Acts for Conversational Recommendation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 1839–1851.

[12] Ahtsham Manzoor and Dietmar Jannach. 2021. Generation-based vs. Retrieval-based Conversational Recommendation: A User-Centric Comparison. In *Fifteenth ACM Conference on Recommender Systems*. 515–520.

[13] Zhaochun Ren, Zhi Tian, Dongdong Li, Pengjie Ren, Liu Yang, Xin Xin, Huasheng Liang, Maarten de Rijke, and Zhumin Chen. 2022. Variational Reasoning about User Preferences for Conversational Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[14] Rajdeep Sarkar, Koustava Goswami, Mihael Arcan, and John Philip McCrae. 2020. Suggest me a movie for tonight: Leveraging Knowledge Graphs for Conversational Recommendation. In *Proceedings of the 28th International Conference on Computational Linguistics*. 4179–4189.

[15] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. 2002. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. 253–260.

[16] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1441–1450.

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

[18] Wilson L Taylor. 1953. "Cloze procedure": A new tool for measuring readability. *Journalism quarterly* 30, 4 (1953), 415–433.

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[20] Lingzhi Wang, Huang Hu, Lei Sha, Can Xu, Kam-Fai Wong, and Daxin Jiang. 2021. Finetuning Large-Scale Pre-trained Language Models for Conversational Recommendation with Knowledge Graph. *arXiv preprint arXiv:2110.07477* (2021).

[21] Xiaoyi Wang, Yaguang Li, and Jie Liu. 2021. Conversation Recommender System Based on Knowledge Graph and Time-series Feature. In *The 5th International Conference on Computer Science and Application Engineering*. 1–5.

[22] Fuzhao Xue, Aixin Sun, Hao Zhang, Jinjie Ni, and Eng Siong Chng. 2022. An Embarrassingly Simple Model for Dialogue Relation Extraction. *arXiv preprint arXiv:2012.13873* (2022).

[23] Hang Yan, Junqi Dai, Xipeng Qiu, Zheng Zhang, et al. 2021. A Unified Generative Framework for Aspect-Based Sentiment Analysis. *arXiv preprint arXiv:2106.04300* (2021).

[24] Bowen Yang, Cong Han, Yu Li, Lei Zuo, and Zhou Yu. 2021. Improving Conversational Recommendation Systems' Quality with Context-Aware Item Meta Information. *arXiv preprint arXiv:2112.08140* (2021).

[25] Tong Zhang, Yong Liu, Peixiang Zhong, Chen Zhang, Hao Wang, and Chunyan Miao. 2021. KECRS: Towards Knowledge-Enriched Conversational Recommendation System. *arXiv preprint arXiv:2105.08261* (2021).

[26] 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*. 177–186.

[27] Jinfeng Zhou, Bo Wang, Ruifang He, and Yuexian Hou. 2021. CRFR: Improving Conversational Recommender Systems via Flexible Fragments Reasoning on Knowledge Graphs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 4324–4334.

[28] Kun Zhou, Wayne Xin Zhao, Shuqing Bian, Yuanhang Zhou, Ji-Rong Wen, and Jingsong Yu. 2020. Improving conversational recommender systems via knowledge graph based semantic fusion. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1006–1014.

[29] Kun Zhou, Wayne Xin Zhao, Hui Wang, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. Leveraging historical interaction data for improving conversational recommender system. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2349–2352.

[30] 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*. 4128–4139.

[31] Jie Zou. 2022. Improving search and recommendation by asking clarifying questions. In *ACM SIGIR Forum*, Vol. 55. 1–2.

[32] 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*. 881–890.

[33] Jie Zou and Evangelos Kanoulas. 2019. Learning to Ask: Question-Based Sequential Bayesian Product Search. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (Beijing, China) *(CIKM '19)*. 369–378.

[34] Jie Zou and Evangelos Kanoulas. 2020. Towards Question-Based High-Recall Information Retrieval: Locating the Last Few Relevant Documents for Technology-Assisted Reviews. *ACM Trans. Inf. Syst.* 38, 3, Article 27 (May 2020), 35 pages.

[35] Jie Zou, Evangelos Kanoulas, and Yiqun Liu. 2020. An Empirical Study on Clarifying Question-Based Systems. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2361–2364.