RevCore: Review-augmented Conversational Recommendation

Yu Lu[†], **Junwei Bao**[‡(✉)], **Yan Song**[†♯], **Zichen Ma**[†],
Shuguang Cui[†♯], **Youzheng Wu**[‡], **Xiaodong He**[‡]
[†]The Chinese University of Hong Kong (Shenzhen)
[‡]JD AI Research [♯]Shenzhen Research Institute of Big Data
[†]{yulu1,zichenma1}@link.cuhk.edu.cn
[‡]{baojunwei,wuyouzheng1,xiaodong.he}@jd.com
[†♯]{songyan,shuguangcui}@cuhk.edu.cn

Abstract

Existing conversational recommendation (CR) systems usually suffer from insufficient item information when conducted on short dialogue history and unfamiliar items. Incorporating external information (e.g., reviews) is a potential solution to alleviate this problem. Given that reviews often provide a rich and detailed user experience on different interests, they are potential ideal resources for providing high-quality recommendations within an informative conversation. In this paper, we design a novel end-to-end framework, namely, Review-augmented Conversational Recommender (**RevCore**), where reviews are seamlessly incorporated to enrich item information and assist in generating both coherent and informative responses. In detail, we extract sentiment-consistent reviews, perform review-enriched and entity-based recommendations for item suggestions, as well as use a review-attentive encoder-decoder for response generation. Experimental results demonstrate the superiority of our approach in yielding better performance on both recommendation and conversation responding.[1]

1 Introduction

With the increasing popularity of intelligent assistants in users' daily lives, how to effectively help users find information or finish specific tasks, such as recommendation and booking, has tremendous commercial potential. Therefore, conversational recommendation (CR) systems have attracted widespread attention for being a tool providing users potential items of interest through dialogue-based interactions. Though existing studies (Sun and Zhang, 2018; Zhang et al., 2018; Lei et al., 2020) proposed to integrate recommender and dialogue components for providing user-specific suggestions through conversations, CR remains challengeable because (i) typical dialogues are short and lack sufficient item information for user preference capturing (Chen et al., 2019; Zhou et al., 2020), and (ii) difficulties exist in generating informative responses with item-related descriptions (Shao et al., 2017; Ghazvininejad et al., 2018; Wang et al., 2019b). Thus, recently, external information in the form of structured knowledge graphs (KG) is introduced to enhance item representations by using rich entity information in KG (Chen et al., 2019; Zhou et al., 2020). While KG-based methods improve CR to some extent, they are still limited in (i) worse versatility resulted from a high cost of KG construction; and (ii) inadequate integration of knowledge and response generation (Lin et al., 2020).

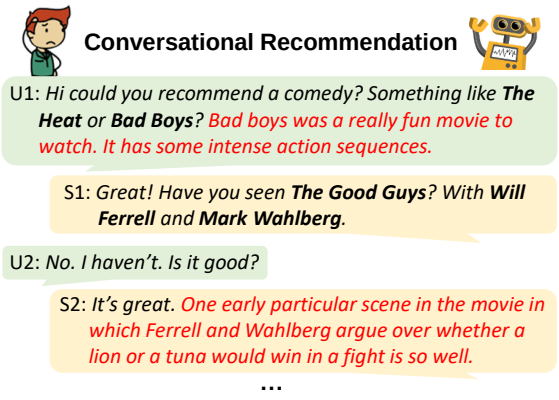

Figure 1: An illustrative example of a user-system conversation on movie recommendation. The additional sentiment-matched reviews are in red. Items (movies) and entities (e.g., actors) are in bold.

Given that, nowadays, users are greatly encouraged to share their consumption experience (e.g., restaurant, traveling, movie, etc.), reviews are easily accessed over the internet. Such reviews often provide rich and detailed user comments on different factors of interest, which are crucial in suggest-

(✉) Corresponding Author

[1]Our code will release in https://github.com/JD-AI-Research-NLP/RevCore.

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 1161–1173
August 1–6, 2021.

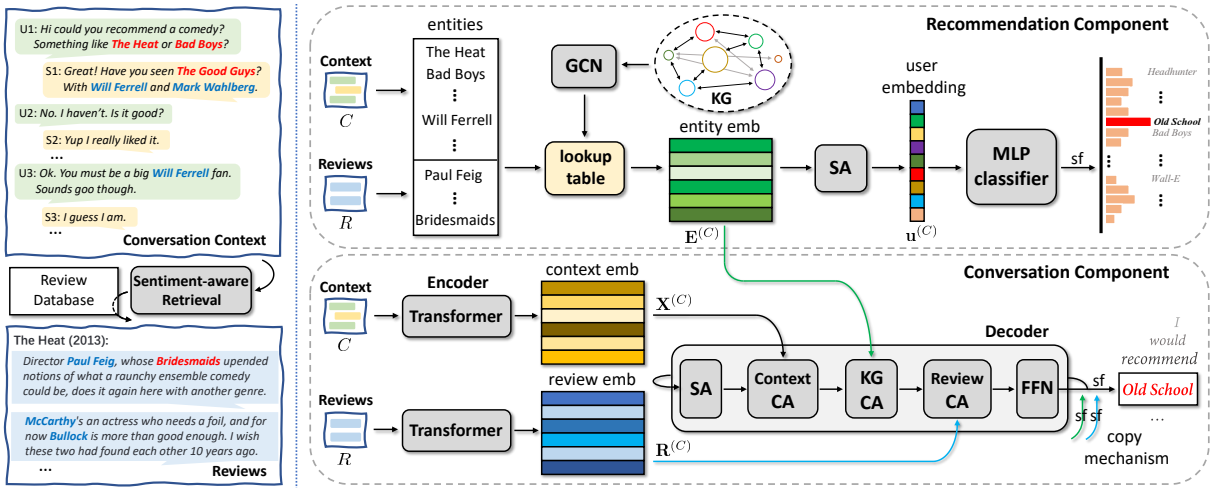

Figure 2: The overview of the proposed method in a movie recommendation scenario, where "emb", "SA", "CA", and "sf" denote embedding, self-attention, cross-attention, and softmax operation, respectively.

ing recommendations to particular users. Thus one can treat reviews as promising external sources for higher-quality recommendations in a conversation. As an example shown in Figure 1, the CR system may be unfamiliar with the mentioned items from the user, resulting in an uninformative response "*It's great.*", thus the chat does not help with recommendation owing to lacking necessary knowledge. In addition, another factor resulting in users' lower acceptance rates to the recommendations is that elaborations on the suggestion are seldom given, which can be alleviated with more explanatory or descriptive utterances after referring to reviews.

Therefore, in better linking external knowledge to recommendation in dialogues, in this paper, we propose a novel framework, Review-augmented Conversational Recommender (**RevCore**), to enhance CR by additional review data. In doing so, we firstly analyze user's utterances with their sentiment polarities and then retrieve reviews for the items mentioned by the user with keeping their sentiment matching the utterances (e.g., they should be both positive or negative). The obtained reviews are thus recommendation-beneficial (He et al., 2015; Hariri et al., 2011) because they are given by the ones who have seen/used and also show interests (or with no interests) in the mentioned items. Afterward, we incorporate the selected reviews into dialogue history, from which the CR system can learn user preference from review-enriched item information. In addition, we also use the sentiment-coordinated reviews to enhance the dialogue response generation, where a review-attentive decoder introduces item information from selected reviews to generate coherent and informative responses. To the best of our knowledge, it is the first time that the aforementioned CR issues have been addressed through incorporating external reviews. Experimental results on a widely used benchmark dataset (Li et al., 2018) show that RevCore is superior on both recommendation accuracy and conversation quality. Further analyses are also performed to confirm the effectiveness of RevCore in an appropriate manner of introducing reviews to CR.

2 The Proposed Framework

We present the proposed Review-augmented Conversational Recommender (**RevCore**) with its overview illustrated in Figure 2, where there are three main components, i.e., the review retrieval module, the recommendation component, and the conversation component. The review retrieval module takes a conversation context C as the input and outputs the selected review set R from the review database $\mathcal{R}_{db}$ which contains all reviews. The context $C = \{s_t\}_{t=1}^{N}$ consists of all utterances s_t of the dialogue history given by the user and system in turns, and the review set R includes all review sentence $r \in \mathcal{R}_{db}$ retrieved according to the contexts in previous turns. With C and R as the input, the recommendation component outputs a set of items from the candidate item set $\mathcal{Z}$ as the recommendation. The dialogue component also accepts C and R as input, and outputs an utterance $s_{t+1} = \{w_i\}_{i=1}^{M}$ as the response, where w_i is the i_{th} word and M the length of s_{t+1}. The output s_{t+1} is added to the context of the next turn.

We first introduce how to retrieve proper reviews

from a database in the following Section 2.1. Then our solutions to the recommendation and conversation tasks are described in Section 2.2 and Section 2.3 respectively, along with detailed illustrations of how reviews enhance both two tasks. Without the loss of generality, our method is introduced in a movie recommendation scenario.

2.1 Review Retrieval

To help dialogue with reviews, given $\mathcal{R}_{db}$, it is of great importance to retrieve proper ones. The reasons are two folds: (i) non-relevant reviews may result in harmful effects to the user representation; (ii) reviews with inconsistent altitudes inject noise into the conversation, which impedes generating coherent responses. Then, a preliminary retrieval is to search in $\mathcal{R}_{db}$ for proper reviews according to the mentioned item in the conversation context C. For review filtering, we design a sentiment-aware retrieval module. The sentiment value $v \in [0, 1]$ of each review r can be captured by a transformer-based sentiment predictor:

$$v = \text{Sentiment}(r),\ r \in \mathcal{R}_{db}, \tag{1}$$

where $\text{Sentiment}(\cdot)$ denotes sentiment prediction, and v can be viewed as how well the movie is liked in this review r. Similarly, the sentiment of a response to this movie can also be obtained in this way. As a result, reviews that possess similar sentiment polarity v^* with the response are selected.

Considering helpful reviews are usually long paragraphs, we only retain part of them, one sentence, for each mentioned movie. Given a context C, there exist two manners to select the sentences $r^{(C)}$ from the raw reviews, word-wisely (or phrase-wisely) and sentence-wisely. The first one randomly chooses some words or phrases to form each "sentence", and one whole sentence is directly selected at random in a second way. Despite the expense of sentence fluency, the first manner enjoys much variability due to the extensive word/phrase combinations. The process to obtain $r^{(C)}$ can be formulated as follows:

$$r^{(C)} = \text{Retrieve}(\mathcal{R}_{db}, V, v^*), \tag{2}$$

where $\text{Retrieve}(\cdot)$ denotes the retrieval operation and V is the set of all v. The obtained $r^{(C)}$ is added into the review set R.

With the retrieved review sentence, one way of incorporation is to briefly insert it right behind the sentence where the item is, as in Fig. 1. However, it may cause the perturbation to the conversational consistency by interrupting the original dialogue. Thus we seamlessly incorporate the review embedding into the conversation component, which is described in Section. 2.3. More importantly, the review sentence serves as a brief introduction or explanation to the mentioned movie. It enriches user information for personalized recommendations and introduces external knowledge for more informative recommendation responses.

2.2 Review-augmented Recommendation

The recommender component is constructed based on a KG-based framework (Zhou et al., 2020), with all entities in the context are extracted to generate the embedding of a user profile. In our method, the retrieved reviews work on enriching entity information so that the user embedding can be augmented to promote recommendation accuracy.

Similar to the approach in Zhou et al. (2020), a candidate entity embedding dictionary $\mathcal{E}$ is constructed first by using GNN to learn entity representations from KG, e.g., DBpedia (Auer et al., 2007). Given a context C, all entities $E^{(C)}$ in it are extracted. Then the embedding vectors of them are looked up from $\mathcal{E}$ and concatenated into a matrix $\mathbf{E}^{(C)} \in \mathbb{R}^{l^{(C)} \times d}$, where $l^{(C)}$ is the number of entities in the context C, and d denotes the embedding dimension. Next, the entity embedding $\mathbf{E}^{(C)}$ is aggregated into a user embedding vector $\mathbf{u}^{(C)}$, through a self-attention layer (SA) as follows:

$$\begin{aligned} \mathbf{u}^{(C)} &= \mathbf{E}^{(C)} \cdot \boldsymbol{\alpha}, \\ \boldsymbol{\alpha} &= \text{softmax}(\mathbf{b}^\top \cdot \tanh(\mathbf{W}_\alpha \mathbf{E}^{(C)})), \end{aligned} \tag{3}$$

where $\boldsymbol{\alpha}$ is the attention weight vector, and $\mathbf{W}_\alpha$ and $\mathbf{b}$ are the parameter matrix and vector for linear projection and bias. Given $\mathbf{u}^{(C)}$, a multi-layer perceptron (MLP) and a softmax operation are adopted to obtain the recommendation prediction $\mathbf{p} \in \mathbb{R}^L$, where L is the number of candidate movies:

$$\mathbf{p} = \text{softmax}(\text{MLP}(\mathbf{u}^{(C)})). \tag{4}$$

To learn parameters in the recommender component, a cross-entropy loss $\mathcal{L}_{rec}$ between the prediction $\mathbf{p}$ and the target movie category is computed:

$$\mathcal{L}_{rec} = -\frac{1}{M} \sum_{i=1}^{M} \log p_i^*, \tag{5}$$

where M is the number of recommendations and p_i^* is the prediction probability of the target category in the i_{th} recommendation.

The pipeline described above suffers from the entity sparsity in dialogue history, resulted from the dataset construction process, where annotators are inevitably unfamiliar with some movies. Retrieved reviews can act to enrich the $E^{(C)}$ by adding more entity words.The process of obtaining review-enriched entities can be formulated as:

$$\begin{aligned} E_C^{(C)} &= \text{Extract}(C), \\ E_R^{(C)} &= \text{Extract}(R^{(C)}), \\ E^{(C)} &= \{E_C^{(C)}, E_R^{(C)}\}, \end{aligned} \tag{6}$$

where $\text{extract}(\cdot)$ defines the entity extraction operation, and $E_c^{(C)}$, $E_r^{(C)}$ denotes entities extracted from the context and retrieved review, respectively. Based on the review-enriched entities, the user embedding is expected to be better represented to produce a more precise recommendation.

2.3 Review-augmented Response Generation

Reviews can also augment the response generation in the conversation component. We build an encoder-decoder framework to handle the generation task. Retrieved reviews and context are encoded separately first, for the purpose of maintaining the dialog consistency. In the decoding stage, the review embedding is fused via an attention layer to generate informative responses. Considering that a good modeling of the input plays an important role to achieve an outstanding model performance (Mikolov et al., 2013; Song and Shi, 2018; Peters et al., 2018; Devlin et al., 2019; Song et al., 2021) and transformer-based approaches have achieved state-of-the-art in many NLP tasks (Vaswani et al., 2017; Chen et al., 2019; Zhou et al., 2020; Chen et al., 2020a; Joshi et al., 2020; Wang et al., 2020; Tian et al., 2020), we adopt two transformers as the encoders for context and reviews. Given a context C and the retrieved reviews R, the context embedding $\mathbf{X}^{(C)}$ and review embedding $\mathbf{R}^{(C)}$ are first obtained:

$$\begin{aligned} \mathbf{X}^{(C)} &= \text{Transformer}_{\boldsymbol{\theta}_X}(C), \\ \mathbf{R}^{(C)} &= \text{Transformer}_{\boldsymbol{\theta}_R}(R), \end{aligned} \tag{7}$$

where $\boldsymbol{\theta}_X, \boldsymbol{\theta}_R$ are parameters in these two transformers. The decoding stage takes them and the entity embedding $\mathbf{E}^{(C)}$ as inputs of attention layers. These attention layers aim to fuse the external information from KG and reviews R into the context information, inspired by the work of Zhou et al. (2020). Given the decoding output of last time unit $\mathbf{Y}^{i-1}$, the current one $\mathbf{Y}^i$ is generated by:

$$\begin{aligned} \mathbf{A}_0^i &= \text{MHA}(\mathbf{Y}^{i-1}, \mathbf{Y}^{i-1}, \mathbf{Y}^{i-1}), \\ \mathbf{A}_1^i &= \text{MHA}(\mathbf{A}_0^i, \mathbf{X}^{(C)}, \mathbf{X}^{(C)}), \\ \mathbf{A}_2^i &= \text{MHA}(\mathbf{A}_1^i, \mathbf{E}^{(C)}, \mathbf{E}^{(C)}), \\ \mathbf{A}_3^i &= \text{MHA}(\mathbf{A}_2^i, \mathbf{R}^{(C)}, \mathbf{R}^{(C)}), \\ \mathbf{Y}^i &= \text{FFN}(\mathbf{A}_3^i), \end{aligned} \tag{8}$$

where $\text{MHA}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$ represents the multi-head attention function (Vaswani et al., 2017), which takes a query, key, and value as input:

$$\begin{aligned} &\text{MHA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = [\text{h}_1, \cdots, \text{h}_h] \cdot \mathbf{W}^o, \\ &\text{h}_i = \text{Attention}(\mathbf{Q}\mathbf{W}_i^{(q)}, \mathbf{K}\mathbf{W}_i^{(k)}, \mathbf{V}\mathbf{W}_i^{(v)}), \end{aligned} \tag{9}$$

where $[\cdot]$ represents the concatenation operation, h is the number of heads, and $\mathbf{W}_i$ is the parameter matrix to learn. $\text{FFN}(\cdot)$ in Equation 8 defines a fully-connected feed-forward network, which comprises of two linear layers with one ReLU activation layer in between:

$$\text{FFN}(\boldsymbol{x}) = \text{ReLU}(\boldsymbol{x}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2. \tag{10}$$

As presented above, information is injected progressively into the decoding stage, from the original context at first, then related entity information in KG, and finally reviews, which contain detailed item-related information.

To complete the generation, the decoder output $\mathbf{Y}^i$ is processed through a softmax operation to predict the token distribution. Apart from the conversational consistency required in the chit-chat task, the CR system also expects recommendation-related responses, which usually contain relevant entities and descriptive keywords. So a copy mechanism is further adapted to introduce vocabulary bias and thus increase the informativeness in the generation. Given the previous generated sub-sequence $\{y_{i-1}\} = y_1, y_2, \cdots, y_{i-1}$, the generation probability y_i of the next token can be computed as:

$$\begin{aligned} \Pr(y_i|\{y_{i-1}\}) = \Pr_1(y_i|\mathbf{Y}_i) + \Pr_2(y_i|\mathbf{Y}_i, G) + \\ \Pr_3(y_i|\mathbf{Y}_i, R), \end{aligned} \tag{11}$$

where $\Pr_1(\cdot)$ is a generation probability function over the vocabulary, with $\mathbf{Y}^i$ as the input. G and R represents the knowledge graph and reviews we use. $\Pr_2(\cdot)$, $\Pr_3(\cdot)$ are copy probability functions from KG entities and reviews, respectively, implemented by a standard copy mechanism (Gulcehre et al., 2016) (computing the distributions over the KG

words or review words). Both probability functions are implemented with a softmax operation. To learn the response generation in the dialogue component, we set a cross-entropy loss:

$$\mathcal{L}_{gen} = -\frac{1}{N}\sum_{t=1}^{N}\log\big(\Pr(s_t|s_1,\cdots,s_{t-1})\big), \quad (12)$$

where N is the number of turns, s_t represents the t_{th} utterance in the conversation.

To train the whole model, it includes three steps: (i) pre-training the sentiment predictor in the review retrieval module; (ii) training the recommender component by minimizing $\mathcal{L}_{rec}$; (iii) training the dialogue component by minimizing $\mathcal{L}_{gen}$.

3 Experiment Settings

3.1 Dataset

REDIAL (Li et al., 2018) is a widely-used dataset of real-world conversations around the theme of providing movie recommendations generated by the human in seeker-recommender pairs. REDIAL contains 10,021 conversations related to 64,362 movies, split into training, validation, and test sets using a ratio of 8:1:1[2]. To construct a review database, we crawled 30 reviews for each movie from IMDb[3] website, which is one of the most popular and authoritative movie databases. Each review can be queried according to the corresponding movie along with its rating and helpful score provided by IMDb. In practice, we select the 30 reviews with the highest helpful scores for each movie to guarantee the high quality of collected reviews. Other manners of selecting the 30 reviews are described and compared in the second part of Section 4.4.

3.2 Implementation Details

The maximum lengths of context and response are set to 256 and 30, respectively. Transformers for review encoding in dialogue generation and sentiment prediction use the same hyper-parameters with the context encoder. For sentiment polarity in the reviews, we threshold on the star-rating to getting sentiment polarity with the threshold set to 5. In the dialogue context, the sentiment polarity is obtained according to users' attitude to the mentioned entity in utterances, which is provided by the REDIAL dataset. Other settings are kept consistent with Zhou et al. (2020) for fair comparison[4]. Besides, the "review sentence" is selected according to the sentiment value and in a sentence-wise manner, and the token number of incorporated review sentences is set to 20, considering the balance between the original source and external source. We add the retrieved review sentences after the mentioned items in the dialogue component training to guide it to generate review-aware responses. The sentiment predictor for reviews is trained on the collected reviews. The sentiment predictor for dialog context is trained on the IMDb Movie Reviews Dataset (Maas et al., 2011) and then finetuned on the REDIAL dataset.

3.3 Baselines

Evaluated on the REDIAL dataset, we compare our approach with a variety of competitive baselines from previous studies listed as follows:

- **Trans** (Vaswani et al., 2017) applies a encoder-decoder framework based on transformer for generation, and applies a transformer encoder to encode context information for recommendation.
- **Redial** (Li et al., 2018) builds a conversation component based on a hierarchical encoder-decoder architecture, and its recommender component is implemented by an auto-encoder extended with a RNN-based sentiment analysis module.
- **KBRD** (Chen et al., 2019) adopts DBpedia-enhanced contextual items or entities to construct user profile for recommendation. The KG-enhanced user profile also serves as word bias for the transformer-based generation module.
- **KGSF** (Zhou et al., 2020) uses MIM (Viola and Wells III, 1997) to align the semantic spaces of two KGs. The user embedding is obtained from the aligned representations of words and items for recommendation. The generation module follows a transformer encoder and a fused KG enhanced decoder.

3.4 Evaluation Metrics

Our method is evaluated on both the recommendation and conversation tasks. The evaluation metric for recommendation is Recall@k (R@k, k = 1, 10, 50), which indicates whether the predicted top-k items contain the ground truth recommendation

[2]More statistics presents in appendix A.

[3]https://www.imdb.com/

[4]More details of hyper-parameters and training strategies are described in Appendix B; the size of different models and their inference speed are reported in Appendix C.

Models	R@1	R@10	R@50
Redial	2.4	14.0	32.0
KBRD	3.1	15.0	33.6
KGSF	3.9	18.3	37.8
RevCore (−KG)	4.2	22.7	43.3
RevCore (+KG)	**6.1**	**23.6**	**45.4**

Table 1: Results on the recommendation task. Best results are in bold.

Models	Dist-2	Dist-3	Dist-4	PPL
Trans	0.148	0.151	0.137	17.0
Redial	0.225	0.236	0.228	28.1
KBRD	0.263	0.368	0.423	17.9
KGSF	0.289	0.434	0.519	**9.8**
RevCore (−KG)	0.373	0.527	**0.615**	10.7
RevCore (+KG)	**0.424**	**0.558**	0.612	10.2

Table 2: Results on the conversation task. Best results are in bold.

provided by human recommenders. Conversation evaluation comprises automatic and human evaluation. The metrics for automatic evaluation are perplexity (PPL) (Jelinek et al., 1977) and distinct n-gram (Dist-n, $n = 2, 3, 4$) (Li et al., 2016). Perplexity is a measurement for the fluency of natural language, where lower perplexity refers to higher fluency. Distinct n-gram is a measurement for the diversity of generated utterances. Specifically, we use distinct 3-gram and 4-gram at the sentence level to evaluate the diversity. The main purpose of our dialog component is a successful recommendation rather than imitating the ground truth responses. Therefore, we provide annotators to manually evaluate the results instead of using BLEU scores. The annotators evaluate the quality of generated dialogue responses from 3 aspects, i.e., coherence, fluency, and informativeness, with each score ranging from 0 to 1.

4 Results and Analysis

4.1 Evaluation on Recommendation Task

For the recommendation task, we adopt Recall@k (R@1, R@10, R@50) for evaluation. As the results summarized in Table 1, our approach outperforms all competitive baselines and achieves 5.9% R@1, 24.0% R@10, and 41.3% R@50, which is the state-of-the-art performance on the REDIAL dataset.[5] Compared with KGSF, RevCore (+KG) achieves significant improvements, with R@1 score improved about 156% (absolutely 2.2), R@50 score improved about 129% (absolutely 4.5), and R@50 score improved about 120% (absolutely 7.6).

We also evaluate the performance of RevCore (−KG), which means the construction of $\mathcal{E}$ removes relation between entities. Instead, an embedding matrix is randomly initialized and learned to represent each entity, without using the GNN-based embedding. In this version, the external knowledge source we introduce is reduced to review only. As the result in the last two rows of Table 1, RevCore (−KG) can achieve competitive results with RevCore (+KG), and outperform KGSF that uses two KGs. According to our observation, although the learning of entity representation is made harder without structured knowledge graphs, the enrichment of dialogue history by reviews makes up the embedding learning. It demonstrates that incorporating reviews is a meaningful method to improve the recommendation in the conversation. We hope this result inspire further research.

4.2 Evaluation on Conversation Task

Automatic Evaluation The results of automatic evaluation on the REDIAL dataset summarize in Table 2. The proposed RevCore outperforms all competitive baselines and achieves significant improvements over most of the automatic metrics. Compared with KGSF, all of the Dist-n scores are significantly lifted, namely, by +0.14 for Dist-2, +0.11 for Dist-3, and +0.08 for Dist-4, which demonstrates our method is effective to generate diverse utterances. Besides, RevCore (+KG) achieves a comparable PPL score with KGSF. It validates our claim that the review incorporation in our method does not cause a decline in generation fluency. The lower PPL score of RevCore (+KG) possibly relates to the high fluency contained in incorporated reviews that carefully induct by website users. For the version of RevCore (−KG), it achieves higher Dist-n scores than KGSF and only results in a slight drop compared with RevCore (+KG). It demonstrates that reviews compared with KG bring more diversity as a richer and more accessible external source.

Human Evaluation We adopt human evaluation on a random selection of 100 multi-turn dialogues

[5] We report the performance of different models on the validation sets in Appendix D and the mean and standard deviation of the test set results in Appendix E.

Models	Coherence	Fluency	Informat
Trans	0.189	0.226	0.115
Redial	0.225	0.455	0.228
KBRD	0.263	0.468	0.283
KGSF	0.324	0.502	0.332
RevCore (−KG)	0.556	0.493	0.682
RevCore (+KG)	**0.601**	**0.567**	**0.718**

Table 3: Human evaluation results. "Informat" denotes informativeness. Best results are in bold.

Models	Dist-2	Dist-3	Dist-4	PPL
RevCore (+KG)	**0.424**	**0.558**	**0.612**	10.2
−revCP	0.353	0.443	0.503	**10.0**
−revRA	0.328	0.428	0.516	13.2
−revEN	0.394	0.534	0.586	10.8

Table 4: Ablation study on the conversation task.

from the testing set. Given one dialogue context, each generated response is scored ranging from 0 to 1, with a higher value indicating a more coherent, fluent, and informative utterance. The final result is calculated as the average score of three annotators, as summarized in Table 3. The proposed RevCore (with or without KG) is consistently better than all the baselines, especially on the metric of informativeness in a large margin. It further proves the effectiveness of our method, and also verifies its superiority in numerical results.

Ablation Study We demonstrate the contribution of each part on the conversation task by constructing an ablation study based on three variants of our complete model, including: (1) RevCore (−revCP) by removing the copy mechanism for reviews, (2) RevCore (−revRA) by removing the review attention layers from the transformer decoder, and (3) RevCore (−revEN) by removing the sentiment-aware review encoder (the reviews share the same encoder with the context). As shown in Table 4, first, all the techniques are useful to improve the final performance in generating diversified utterances. Besides, the copy mechanism and the review attention layers seem to be more important in conversation diversity. One of the potential reasons is that these two components are directly related to the decoding stage. Separated encoders for review and context lead to a slight increment, which shows that sharing a common encoder is an alternative solution.

Len	Recommendation			Conversation		
	R@1	R@10	R@50	Dist-3	Dist-4	PPL
10	4.5	21.8	37.0	0.491	0.564	**10.2**
20	**6.1**	**23.6**	**45.4**	**0.558**	**0.612**	**10.2**
30	4.7	22.0	41.3	0.263	0.426	13.1
40	5.8	21.3	41.3	0.289	0.439	14.0
50	5.1	22.3	40.8	0.304	0.520	13.8

Table 5: Performance of RevCore when incorporating review sentences with different length (Len). Best results are in bold.

4.3 Case Study

In this part, we present a visualized example to illustrate how our model works in practice, as shown in Figure 3. First, the sentiment-aware review retrieval module guarantees the coherence of incorporating reviews to some extent, for example in Figure 3, negative comments (the last row for the movie *The Notebook*) are filtered out considering the positive attitude in the original utterance. Secondly, incorporated reviews exactly enrich the context for better recommendation. As seen in the first yellow frame, many entities like "*Roshan Andrews*" mentioned in the review are added into the entity set. Note that some other entities are also added from the reviews that are incorporated into users' utterances as described in Section 2, which is not visualized here but brings recommendation accuracy improvement as well. Last but not least, the generated responses are more informative to use more varied expressions like "*the magic spell*" and "*the sultry dance*". Besides, they uncover more details related to the movie that acts as explanatory sentences that make recommendations accepted more easily and naturally.

4.4 Discussion

Longer Review, Better Performance? In our basic setting, each retrieved review sentence is formed by 20 words. We conduct a series of experiments by setting the length of retrieved review sentences as 10, 20, 30, 40, and 50 to inspect the effect of review length. The results of using different lengths are shown in Table 5, among which 20 is the best across all metrics. An interesting finding is that continually increasing the review length makes no benefits after reaching 20. Over introducing external text may suppress original text, thus 20 is a better choice to keep the balance between them.

Figure 3: Case study. $U(i)$ (green) and $S(i)$ (yellow) represent user and system, respectively. In the "Dialogue Generation", items are marked in blue font, explanatory sentences are in red. Items are in bold font in "Entities" frames. In "Recommend" frames, a darker color represents a higher probability. "Review Retrieval" gives retrieved review examples, with their sentiment value (0-9) at the most left, and the selected reviews are in bold.

Method	Recommendation			Conversation		
	R@1	**R@10**	**R@50**	**Dist-3**	**Dist-4**	**PPL**
iCorpus	2.1	8.5	20.3	0.430	0.555	11.2
R-H-W	2.3	10.7	25.2	0.307	0.347	10.9
C-H-W	5.4	23.3	42.9	0.471	0.559	13.1
C-H-S	4.2	22.4	39.4	0.534	0.586	11.4
C-S-S	**6.1**	**23.6**	**45.4**	**0.558**	**0.612**	**10.2**

Table 6: Results of various retrieval strategies. Three characters from left to right represent three factors: (i) C/R: correctly/randomly matched movie-review pairs; (ii) S/H: ranking by sentiment value or helpful score respectively; (iii) S/W: sentence/word-wise manner. "iCorpus" indicates using irrelevant corpus. Best results are in bold.

Appropriate Reviews Help More? The "review sentence" are obtained from a 3-stage process, namely, searching item-matched reviews from the database, ranking them by the helpful score or sentiment value, and constructing "review sentence" word-wisely or sentence-wisely. Therefore, we conduct control experiments to inspect these three factors. As shown in Table 6, (i) using reviews randomly matched with items (R-H-W) results in significantly lower R@k and Dist-n scores; (ii) ranking by sentiment value (C-S-S) leads to better performance across all metrics than by helpful score (C-H-S), which demonstrates the necessity of using sentiment-aware review retrieval; (iii) the sentence-wise manner (C-H-S) gets a lower PPL than the word-wise one (C-H-W), which is reasonable because the incorporated reviews made up of random words causes the fluency loss. Besides, another experiment is conducted to verify the necessity of using a movie-review database. A food-review database is constructed as a topic-irrelevant corpus (iCorpus), which results in the lowest R@k yet not bad Dist-n scores. It shows that despite the response diversity brought by the external corpus, the unrelated entities from another domain have negative impacts on the recommendation accuracy.

5 Related Work

Recommender systems have emerged as a separate research area and now play an indispensable role in daily social lives. Traditional recommender systems tend to work statically, primarily relying on content-based approaches or the collaborative filtering hypothesis (Resnick et al., 1994; Pazzani and Billsus, 2007; Wang et al., 2019b), which assumes that similar users may have similar interests. Afterward, more sophisticated methods using neural networks are proposed and prove effective. For instance, neural factorization machines (He and Chua, 2017) and deep interest networks (Zhou et al., 2018) are used to estimate user preferences based on historical user-item interactions. Graphs are adopted in Wang et al. (2019b,a) to model complex relations among users, items, and attributes for a better representation of data.

In recent years, major advances made in dialog systems (Dodge et al., 2016; Yan et al., 2016; Benni et al., 2016; Bordes et al., 2017; Song et al., 2020) and structured knowledge-based info-seeking technics including question answering (Bao et al., 2014,

2016; Yin et al., 2015; Yih et al., 2015; Shao et al., 2019) and question generation (Serban et al., 2016; Bao et al., 2018; Dušek et al., 2020) have encouraged the development of *conversational recommendation* systems, which dynamically obtain user preferences through interactive conversation with users. Multiple datasets have been constructed (Dodge et al., 2016; Li et al., 2018; Kang et al., 2019) to facilitate the study of this task. Li et al. (2018) collect a standard human-to-human multi-turn dialog dataset focusing on providing movie recommendations. Based on these datasets, various approaches are proposed to address different issues in CR systems. Specifically, external information is introduced to alleviate the cold-start problem, including knowledge bases (Wang et al., 2018), social networks (Daramola et al.), and knowledge graphs (Chen et al., 2019). Christakopoulou et al. (2016) use bandit-based explore-exploit strategy to minimize the number of user queries. Liu et al. (2020) conduct multi-goal planning to make a proactive conversational recommendation over multi-type dialogues. A multi-view method is proposed in Chen et al. (2020b) for the explainable conversational recommendation. The work of Pecune et al. (2020) builds a socially aware CR system engaging its users through a rapport-building dialogue to improve users' perception.

Different from all aforementioned previous work, we offer an alternative to AIG with an augmented conversational recommendation system by incorporating reviews that highly relevant to items. Particularly, our model is able to learn better user representations from a review-enriched dialogue context, which enables a high-quality recommendation and response generation.

6 Conclusion

In this paper, we proposed a novel CR framework with review augmentation, including a sentiment-aware retrieval module, a recommender exploiting the review-enriched user profile, an encoder for enhancing semantic embedding of selected reviews, and a review attentive decoder to integrate review information for dialogue response generation. Experimental results show that our approach achieves consistent and significant improvements of both recommendation and dialogue responding over baselines, and is able to generate informative responses without losing fluency and coherence.

Acknowledgement

The work was supported in part by the Key Area R&D Program of Guangdong Province with grant No. 2018B030338001, by the National Key R&D Program of China with grant No. 2018YFB1800800, by Shenzhen Outstanding Talents Training Fund, by Guangdong Research Project No. 2017ZT07X152, and by the National Key Research and Development Program of China under Grant No. 2018YFB2100802.

References

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer.

Junwei Bao, Nan Duan, Zhao Yan, Ming Zhou, and Tiejun Zhao. 2016. Constraint-based question answering with knowledge graph. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2503–2514, Osaka, Japan. The COLING 2016 Organizing Committee.

Junwei Bao, Nan Duan, Ming Zhou, and Tiejun Zhao. 2014. Knowledge-based question answering as machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 967–976, Baltimore, Maryland. Association for Computational Linguistics.

Junwei Bao, Duyu Tang, Nan Duan, Zhao Yan, Yuanhua Lv, Ming Zhou, and Tiejun Zhao. 2018. Table-to-text: Describing table region with natural language. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5020–5027. AAAI Press.

Dominique Benni, Francis Chauvet, and Alain Guillot. 2016. Human-machine dialog system. US Patent 9,411,370.

Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2017. Learning end-to-end goal-oriented dialog. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Guimin Chen, Yuanhe Tian, and Yan Song. 2020a. Joint aspect extraction and sentiment analysis with directional graph convolutional networks. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 272–279,

Barcelona, Spain (Online). International Committee on Computational Linguistics.

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)*, pages 1803–1813, Hong Kong, China. Association for Computational Linguistics.

Zhongxia Chen, Xiting Wang, Xing Xie, Mehul Parsana, Akshay Soni, Xiang Ao, and Enhong Chen. 2020b. Towards explainable conversational recommendation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 2994–3000. ijcai.org.

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, San Francisco, CA, USA, August 13-17, 2016*, pages 815–824. ACM.

Olawande Daramola, Gleb Sizov, and Pinar Öztürk. Improving trust in travel recommendations using a conversational textual cbr framework.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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 *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2020. Evaluating the state-of-the-art of end-to-end natural language generation: The e2e nlg challenge. *Computer Speech & Language*, 59:123–156.

Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. 2018. A knowledge-grounded neural conversation model. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5110–5117. AAAI Press.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 140–149, Berlin, Germany. Association for Computational Linguistics.

Negar Hariri, Bamshad Mobasher, Robin Burke, and Yong Zheng. 2011. Context-aware recommendation based on review mining. In *ITWP@ IJCAI*.

Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. 2015. Trirank: Review-aware explainable recommendation by modeling aspects. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015*, pages 1661–1670. ACM.

Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, pages 355–364. ACM.

Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker. 1977. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63–S63.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

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)*, pages 1951–1961, Hong Kong, China. Association for Computational Linguistics.

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 *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, pages 304–312. ACM.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*,

pages 110–119, San Diego, California. Association for Computational Linguistics.

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 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 9748–9758.

Xiexiong Lin, Weiyu Jian, Jianshan He, Taifeng Wang, and Wei Chu. 2020. Generating informative conversational response using recurrent knowledge-interaction and knowledge-copy. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 41–52, Online. Association for Computational Linguistics.

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*, pages 1036–1049, Online. Association for Computational Linguistics.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781*.

Michael J Pazzani and Daniel Billsus. 2007. Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer.

Florian Pecune, Lucile Callebert, and Stacy Marsella. 2020. A socially-aware conversational recommender system for personalized recipe recommendations. In *Proceedings of the 8th International Conference on Human-Agent Interaction*, pages 78–86.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. 1994. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186.

Iulian Vlad Serban, Alberto García-Durán, Caglar Gulcehre, Sungjin Ahn, Sarath Chandar, Aaron Courville, and Yoshua Bengio. 2016. Generating factoid questions with recurrent neural networks: The 30M factoid question-answer corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 588–598, Berlin, Germany. Association for Computational Linguistics.

Bo Shao, Yeyun Gong, Junwei Bao, Jianshu Ji, Guihong Cao, Xiaola Lin, and Nan Duan. 2019. Weakly supervised multi-task learning for semantic parsing. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 3375–3381. ijcai.org.

Yuanlong Shao, Stephan Gouws, Denny Britz, Anna Goldie, Brian Strope, and Ray Kurzweil. 2017. Generating high-quality and informative conversation responses with sequence-to-sequence models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2210–2219, Copenhagen, Denmark. Association for Computational Linguistics.

Yan Song and Shuming Shi. 2018. Complementary learning of word embeddings. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 4368–4374. ijcai.org.

Yan Song, Yuanhe Tian, Nan Wang, and Fei Xia. 2020. Summarizing medical conversations via identifying important utterances. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 717–729, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yan Song, Tong Zhang, Yonggang Wang, and Kai-Fu Lee. 2021. ZEN 2.0: Continue Training and Adaption for N-gram Enhanced Text Encoders. *arXiv preprint arXiv:2105.01279*.

Yueming Sun and Yi Zhang. 2018. Conversational recommender system. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pages 235–244. ACM.

Yuanhe Tian, Yan Song, and Fei Xia. 2020. Supertagging Combinatory Categorial Grammar with attentive graph convolutional networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6037–6044, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Paul Viola and William M Wells III. 1997. Alignment by maximization of mutual information. *International journal of computer vision*, 24(2):137–154.

Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, pages 417–426. ACM.

Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019a. Neural graph collaborative filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, pages 165–174. ACM.

Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. 2019b. Explainable reasoning over knowledge graphs for recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5329–5336.

Yingyao Wang, Junwei Bao, Guangyi Liu, Youzheng Wu, Xiaodong He, Bowen Zhou, and Tiejun Zhao. 2020. Learning to decouple relations: Few-shot relation classification with entity-guided attention and confusion-aware training. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5799–5809, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Zhao Yan, Nan Duan, Junwei Bao, Peng Chen, Ming Zhou, Zhoujun Li, and Jianshe Zhou. 2016. DocChat: An information retrieval approach for chatbot engines using unstructured documents. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 516–525, Berlin, Germany. Association for Computational Linguistics.

Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1321–1331, Beijing, China. Association for Computational Linguistics.

Pengcheng Yin, Nan Duan, Ben Kao, Junwei Bao, and Ming Zhou. 2015. Answering questions with complex semantic constraints on open knowledge bases. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015*, pages 1301–1310. ACM.

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 2018, Torino, Italy, October 22-26, 2018*, pages 177–186. ACM.

Guorui Zhou, Xiaoqiang Zhu, Chengru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pages 1059–1068. ACM.

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 *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 1006–1014. ACM.

A Statistics for Conversation Dataset and Reviews

Conversations in the REDIAL dataset consist of 163,820 utterances, of which 15.80% have reviews added. The vocabulary size of REDIAL is increased by 13.14% (from 23,356 to 26,427). Among the mentioned 6,927 movies in all conversations, 40% of them are randomly chosen and linked with reviews to keep the balance between the original source and external source. We count the ratio of "disliked" movies by the recommender to explain the improvements brought by doing sentiment-aware retrieval when incorporating reviews. We also show the ratio of unseen movies by the recommender to show the need of introducing reviews to "talk more". Comprehensive statistics are listed in Table 7.

B Experiment Details

Hyper-parameter Settings For a fair comparison, most hyper-parameters are kept consistent with KGSF. We did not search for more hyper-parameters combinations to achieve additional improvements apart from our main idea. The shared hyper-parameters include: embedding dimension

Items	Statistic	Items	Statistic
Dialogues	10,021	Cnd Movies	64,368
Utterances	163,820	Mnt Movies	6,924
+reviews	25,884	+reviews	2,711
Disliked	4.9%	Not Seen	31.9%
Liked	81.2%	Seen	61.3%
Did not say	13.9%	Did not say	6.8%

Table 7: Statistics for the REDIAL dataset with incorporated reviews. "Cnd" denotes "Candidate", and "Mnt" denotes "Mentioned" to indicate the movies that mentioned in the conversations.

Models	Param	Tra Time	Inf Time
KGSF	130.51	4979.88	65.30
RevCore (−KG)	71.12	1045.65	59.57
RevCore (+KG)	133.30	3308.28	51.46

Table 8: Comparison of three models on the number of parameters (million), training ("Tra") time for 30 epochs (second), and inference ("Inf") time (second).

set as 128 in the recommender component and 300 in the dialogue component, the layer number of both GNN in the KG module as 1, the batch size as 32, word embedding initialization via word2vec[6], the optimizer as Adam, the learning rate as 0.001, the epoch number as 30, etc.

Training Strategies To train the whole model, three steps are included: (i) pre-training the sentiment predictor in the review retrieval module; (ii) training the recommender component by minimizing $\mathcal{L}_{rec}$; (iii) training the dialogue component by minimizing $\mathcal{L}_{gen}$. In the first step, the predictor takes each sentence in the review as input and outputs the sentiment, with the corresponding rating set as the label. In the second and third steps, our implementation refers to the training algorithm for the KGSF model. It first pre-trains the parameters in KG for entity representation by minimizing the Mutual Information Maximization loss between two KG embedding, then trains the recommender component by minimizing the recommendation loss and also updating the parameters in the KG module, and finally the dialogue component by minimizing the generation loss with all other modules' parameters "frozen".

[6] https://radimrehurek.com/gensim/models/word2vec.html

Metrics	RevCore (+KG)		RevCore (−KG)	
	Val	Test	Val	Test
R@1	6.13	6.11	4.55	4.19
R@10	23.49	23.62	23.35	22.71
R@50	40.65	45.43	45.33	43.28
Dist-2	0.418	0.424	0.410	0.373
Dist-3	0.582	0.558	0.582	0.527
Dist-4	0.675	0.612	0.668	0.615
PPL	10.89	10.24	10.14	10.69

Table 9: Validation (Val) and test results on the REDIAL dataset of RevCore (+KG) and RevCore (−KG).

Metrics	RevCore (+KG)		RevCore (−KG)	
	Mean	Devi	Mean	Devi
R@1	5.70	± 0.67	3.75	± 0.13
R@10	22.80	± 1.81	21.53	± 0.68
R@50	40.75	± 2.18	44.68	± 0.43
Dist-2	0.394	± 0.039	0.373	± 0.031
Dist-3	0.551	± 0.062	0.527	± 0.051
Dist-4	0.633	± 0.073	0.616	± 0.062

Table 10: Mean and deviation of recall rates (%) and distance scores for RevCore (+KG) and RevCore (−KG).

C Model Size and Running Speed

The model size and running speed of KGSF, RevCore (+KG), and RevCore (−KG) are all listed in Table 8. Note that all three models are implemented with Pytorch[7], trained for 30 epochs, and experimented on NVIDIA A100-SXM4 for 5 times to compute the average running time.

D Results on the Validation Set

We present the validation result of RevCore with and without KG on the REDIAL dataset as a reference for reproducing. All validation results are shown in Table 9, with test results as well.

E Mean and Standard Deviation

We implement the major experiment 4 times to inspect the mean and standard deviation of the performance of RevCore across all metrics. The reported results in the paper of both recommendation accuracy and conversation quality are the mean results. Results are shown in Table 10.

[7] https://pytorch.org/