BARCOR: Towards A Unified Framework for Conversational Recommendation

Anonymous ACL submission

Abstract

Recommendation systems focus on helping users find items of interest in the situations of information overload, where users' preferences are typically estimated by past observed behaviors. In contrast, conversational recommendation systems (CRS) aim to understand users' preferences via interactions in conversation flows. CRS is a complex problem that consists of two main tasks: (1) recommendation and (2) response generation. Previous work often tried to solve the problem in a modular manner, where recommenders and response generators are separate neural models. Such modular architectures often come with a complicated and unintuitive connection between the modules, leading to inefficient learning and other issues. In this work, we propose a unified framework based on BART for conversational recommendation, which tackles two tasks in a single model. Furthermore, we also design and collect a lightweight knowledge graph for CRS in the movie domain. The experimental results show that the proposed methods achieve the state-of-the-art. [1]

1 Introduction

Though recommendation systems have gained tremendous success in various domains and many aspects of our lives, they have potential limitations. Practically, recommending is often a one-shot, reactive, uni-directional process. Users passively receive recommended information from the systems in certain pre-defined situations. It assumes that a user has clear, immediate requests when interacting with the system; however, such recommending may not be accurate since user demand would change over time and vacillate. Sometimes users are indecisive; to this end, traditional recommendation systems lack proactive guidance. Conversational Recommendation Systems (CRS) became an emerging research topic, focusing on exploring users' preferences through natural language interaction. Generally speaking, CRSs support goal-oriented, multi-turn dialogues, which proactively acquire precise user demand by interactions. Thereby, CRS is a complex system consisting of a recommendation module and a dialogue module, which make suitable recommendations and generate proper responses respectively.

In terms of modeling, CRS requires seamless integration between the recommendation module and the dialogue module. The systems need to understand user preferences by preceding dialogue context and recommend suitable items. To recommend items to users in the natural language form, the generated responses need to contain relevant items while being fluent and grammatically correct. Previous work has proposed different approaches for integrating the two major modules, for instance, building belief trackers over semi-structured user queries (Sun and Zhang, 2018; Zhang et al., 2020) and switching decoders for component selection (Li et al., 2018). Furthermore, as practical goal-oriented dialogue systems, CRSs usually utilize Knowledge Graphs (KG) for introducing external knowledge and system scalability. Choosing a suitable KG, leveraging the information of entities, and interacting with the two main components of CRS for high-quality recommendation is undoubtedly another challenging problem.

Recent work (Zhou et al., 2020) proposed to incorporate two special KGs for enhancing data representations of both components and fuse the two semantic spaces by associating two different KGs. Specifically, they incorporate ConceptNet (Speer et al., 2017) for word-level information and DBpedia (Lehmann et al., 2015) for item information. ConceptNet provides word information such as synonyms and antonyms of certain words, which helps understand dialogue context. At the same time, DBpedia has structural information of entities, providing rich attributes and direct relations between

[1] The data and source code will be released once accepted.

items. However, these public large-scale knowledge graphs were not designed for CRSs hence may not be suitable. Though prior methods have achieved some improvement in performance, there are some potential limitations. Most of them build recommender and response generator separately with complicated and unintuitive connection between the modules, which may cause inefficient learning and unclear knowledge transfer between the modules. For example, the work mentioned above (Zhou et al., 2020) requires training multiple graph convolution networks for KG embeddings, mutual information maximization to bridge the embedding spaces. In this case, the practical usage and scalability of the system design are a concern to some extent.

To this end, we propose a unified framework for the conversational recommendation, which tackles two tasks in a single model. The framework is built on top of pretrained BART (Lewis et al., 2020) and finetuned on the recommendation and response generation tasks. We proposed to use the bidirectional encoder of BART as the recommender and the auto-regresive decoder as the response generator, so-called **BARCOR** (**B**idirectional **A**uto-**R**egressive **CO**nversational **R**ecommender). Moreover, we design and collect a lightweight knowledge graph for CRS in the movie domain. With the essentially-connected model structure of BART, we do not need to worry about designing a connection between the recommender and the response generator.

To sum up, the contributions can be summarized as 3-fold:

- This paper proposes a general framework conversational recommendation based on BART, which tackles two tasks in a single model.
- This work designs and collects a lightweight knowledge graph for CRS in the movie domain.
- The benchmark experiments demonstrate the effectiveness of the proposed framework.

2 Related Work

As a specific type of goal-oriented dialogue systems, Conversational Recommendation Systems (CRS) have also moved towards the use of neural networks (Li et al., 2018). Christakopoulou et al. (2018) uses recurrent neural network-based models to recommend videos to users; Zhang et al. (2016) explores the use of knowledge bases in recommendation tasks. Sun et al. (2018) proposes an embedding-based approach to learn semantic representations of entities and paths in a KG to characterize user preferences towards items. Wang et al. (2019) improves the performance of the recommenders by learning the embeddings for entities in the KG using the TransR algorithm (Lin et al., 2015) and refining and discriminating the node embeddings by using attention over the neighbour nodes of a given node. Wang et al. (2018) and Li et al. (2020) focus on solving the task of goal-oriented conversation recommendation for cold-start users. Li et al. (2020) generates new venues for recommendation using graph convolution networks (GCNs) and encodes the dialogue contents using hierarchical recurrent encoder-decoder (HRED) (Sordoni et al., 2015) and thereby recommend locations to users. Li et al. (2018) released the ReDial dataset wherein users are recommended movies based on the conversation they have with the recommendation agents. KBRD (Chen et al., 2019) extends the work of Li et al. (2018) by incorporating a KG and proposing a graph-based recommender for movie recommendations. They have also shown that dialogue and recommendation in CRSs are complementary tasks and benefit one another. To better understand user's preferences, KGSF (Zhou et al., 2020) introduces a word-oriented KG to facilitate node representation learning. Recently, to generate natural and informative responses with accurate recommendations, Lu et al. (2021) incorporates movie reviews, and Zhang et al. (2021) proposes supervision signals for the semantic fusion of words and entities.

3 Dataset

The ReDial (Li et al., 2018) dataset is widely adopted for the conversational recommendation task. This dataset is constructed through Amazon Mechanical Turk (AMT) and comprises multi-turn conversations centered around movie recommendations in seeker-recommender pairs. It contains 10,006 conversations consisting of 182,150 utterances related to 51,699 movies.

To generate training data, previous work (Zhou et al., 2020) viewed all items mentioned by recommenders as recommendations. However, this processing measure causes issues, clearly stated in Zhang et al. (2021). First, repetitive items are likely to guide a model to simply recommend items once appeared in dialogues. Secondly, the evaluation

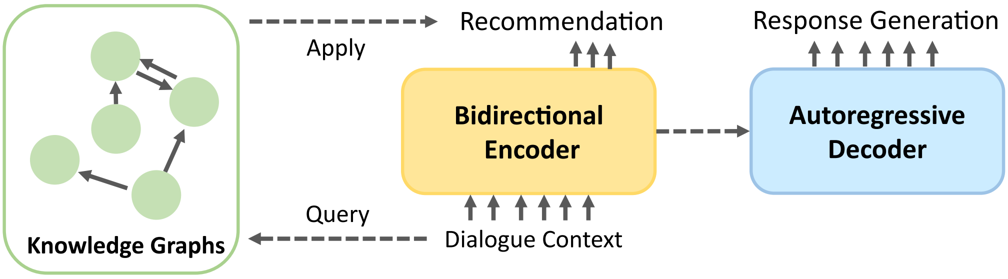

Figure 1: The proposed framework is composed of three components: (1) knowledge graphs for providing external knowledge, (2) a bidirectional encoder as the recommender, and (3) an auto-regressive decoder as the response generator.

dataset is biased to repetitive recommendations, failing to present recommendation quality faithfully. To address the issues, we only consider items as recommendations only if they aren't mentioned before.

Since the recommendation module takes over the item recommendation task, the dialogue module could focus on capturing sentence semantics to produce fluent conversations. Thus, we mask the recommended items in the target response with a special token, `[MOVIE]`. It also serves as a placeholder for items retrieved by the recommender module in generated responses during the inference phase. Table 1 shows training examples from this process.

4 Preliminaries

In this section, we first introduce the problem formulation and then detail the collected knowledge graph.

4.1 Problem Formulation

For the dataset, $\{u_i\}^n$ denotes a conversation, where u_i is the utterance at i-th turn, and n is the number of conversation history. We process a conversation into multiple data triplets $(X, \mathcal{I}, y)$. At j-th turn, $X_j = \{u_i\}_{i=1}^{j-1}$ denotes the conversation context, $\mathcal{I}_j$ is the set of ground truth items presented in u_j for the recommendation task, and $y_j = u_j$ denotes the target response for the generation task. Note that every entry in $\mathcal{I}_j$ cannot appear in the context X_j as stated in the previous section, and it can be an empty set when there is no need for recommendations. For the knowledge graph, $\mathcal{G} = \{(e_h, r, e_t) | e_h, e_t \in \mathcal{E}, r \in \mathcal{R}\}$ denotes the KG, where (e_h, r, e_t) means the head entity e_h and the tail entity e_t is related by the relation r. The entity set $\mathcal{E}$ consists of a movie item set $\mathcal{I}$ and a set of descriptive entities that are film properties. The set of ground truth items $\mathcal{I}_j$ is the subset of $\mathcal{I}$.

The conversational recommendation is essentially the combination of two tasks: document retrieval and natural language generation. They are formulated as two objective functions, $f(X, \mathcal{G})$ and $g(X, \mathcal{I}_{\text{pred}})$. $f(X, \mathcal{G})$ gives novel recommendations $\mathcal{I}_{\text{pred}}$ based on the context X and the KG $\mathcal{G}$, and $g(X, \mathcal{I}_{\text{pred}})$ generates natural responses based on the context and the recommended items.

4.2 CORG (COnversational Recommender Graphs)

In previous works, a wide variety of external knowledge sources are incorporated to facilitate recommendations. However, the KGs adopted in the previous work (Zhou et al., 2020; Chen et al., 2019; Sarkar et al., 2020) are open-domain KGs, e.g., DBpedia and ConceptNet, which may introduce too many irrelevant entities and obscure high-order connectivity as stated in Zhang et al. (2021). Although some datasets, MindReader (Brams et al., 2020) is intended for movie recommendations, its coverage of movies in the ReDial dataset is low, as shown in Table 2. To mitigate these issues, we construct a knowledge graph called **CORG** (**CO**nversational **R**ecommender **G**raphs), which contains 5 types of node entities and 5 types of relations.

Data Source We collect information of movies from Wikidata[2], which is a collaboratively edited multilingual knowledge graph hosted by Wikimedia Foundation[3]. It contains movie-related information and identifiers of other databases for additional information, such as synopses or reviews.

[2] https://www.wikidata.org/wiki/Wikidata:Main_Page
[3] https://wikimediafoundation.org/

	Accepted	Context	Response	Target movie
(a)	✗	S: Hi, I am looking for a movie like Super Troopers. R: You should watch **Police Academy**. S: Is that a great one? I have never seen it.	Yes `[MOVIE]` is funny.	**Police Academy**
(b)	✓	R: Hello, what kind of movies do you like? S: I am looking for a movie recommendation. S: When I was younger, I really enjoyed the A Nightmare on Elm Street.	Oh, you like scary movies? I recently watched `[MOVIE]`.	**Happy Death Day**

Table 1: Examples in the processed ReDial dataset. In the column of context, "S" and "R" represent a movie seeker and a recommender respectively. Recommended items in responses are masked by `[MOVIE]`. Example (a) isn't accepted to the processed dataset since "Police Academy" is a repetitive item, which is presented in the context.

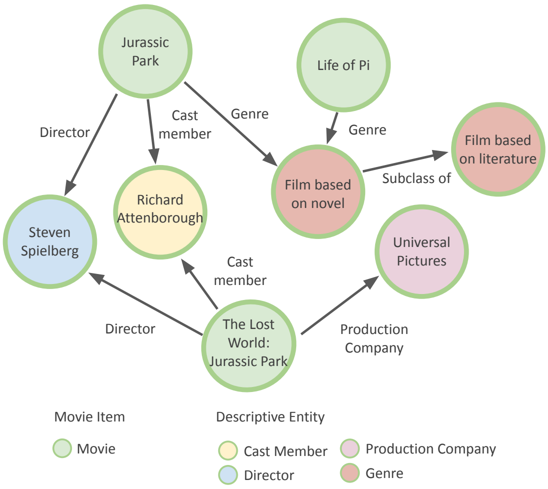

Figure 2: A sample subgraph of CORG. CORG has 5 types of node entities and 5 types of relations, the statistics of types and relations are shown in Table 4.

Information Collection Nodes in CORG comprise two kinds of entities: *movies items* and *descriptive entities*. Movies items are all mentioned movies in ReDial, and descriptive entities are associative properties of those movies. We use "movie name" and "release year" as keys to query Wikidata to collect movie properties, including movie genres, cast members, directors, and production companies. In this way, we get the entire set of nodes in CORG, whose statistics are shown in Table 4. Among 6,924 mentioned movies in ReDial, CORG covers 6,905 movies (99.7%).

Data Processing Assuming seekers are only interested in protagonists, we select top-10 main cast members. Besides, since movie genres in Wikidata are hierarchically arranged (e.g, superhero film is a subclass of action and adventure films), we recursively build edges between the nodes of genres and those of their parent genres. The edge statistics are shown in Table 4.

5 BARCOR

We propose to use the bidirectional encoder of BART (Lewis et al., 2020) as the recommender and the auto-regresive decoder as the response generator, so-called **BARCOR** (**B**idirectional **A**uto-**R**egressive **CO**nversational **R**ecommender). BARCOR is a unified framework for the conversational recommendation which tackles two tasks in a single model. The proposed framework is composed of three main components: (1) a knowledge graph encoder to provide external knowledge, (2) a bidirectional encoder for recommendation, and (3) an auto-regressive decoder for response generation. In this section, we will go through the design of each component in the pipeline.

5.1 Graph Encoder

We follow Zhou et al. (2020), adopting Relational Graph Convolutional Network (R-GCN) (Schlichtkrull et al., 2017) to learn entity representations in CORG. Formally, the hidden state of an entity i at the $(l+1)$-th layer is formulated as:

$$\mathbf{h}_i^{(l+1)} = \sigma\Big(\sum_{r\in\mathcal{R}}\sum_{j\in\mathcal{E}_i^r}\frac{1}{|\mathcal{E}_i^r|}\mathbf{W}_r^{(l)}\mathbf{h}_j^{(l)} + \mathbf{W}^{(l)}\mathbf{h}_i^{(l)}\Big),$$

where $\mathbf{h}_i^{(l)} \in \mathbb{R}^{d_E}$ is the hidden state of the entity i at the l-th layer, d_E is the dimension of the hidden state, and $\mathbf{h}_i^0$ is also referred to as the entity embedding $\mathbf{e}_i$. $\mathcal{E}_i^r$ is the set of neighboring entities of the entity i related by r, and its cardinality serves as a normalization constant. $\mathbf{W}_r^{(l)}$ denotes a learnable relation-specific transformation matrix for the hidden states of neighboring entities under a relation r, and $\mathbf{W}^{(l)}$ is a learnable matrix for transforming hidden states at the l-th layer. We treat the hidden states of the last layer as the representations of entities in CORG, which is denoted by $\mathbf{H} \in \mathbb{R}^{(|\mathcal{E}|\times d_E)}$. The representations construct a search space of recommended candidates for item retrieval.

Knowledge Graph	# Movies	# Entities	Designed for ReDial	Movie Coverage for ReDial
MinderReader (Brams et al., 2020)	4,941	18,707	✗	44.6%
DBpedia (KGSF) (Zhou et al., 2020)	6,111	64,361	✓	88.2%
TMDKG (Zhang et al., 2021)	6,692	15,822	✓	96.2%
CORG	**6,905**	**23,164**	✓	**99.7%**

Table 2: Characteristics of CORG and existing knowledge graphs. Although TMDKG has high movie coverage, their source code is not publicly available.

Other than the recommendation task, we include the node classification task to facilitate graph representation learning. Given an entity representation $\mathbf{h}$ and a multiple layer perceptron (MLP), we obtain a node type prediction $\mathbf{p}_{\text{node}} \in \mathbb{R}^{N_T}$, where N_T is the number of node types:

$$\mathbf{p}_{\text{node}} = \text{Softmax}(\text{MLP}(\mathbf{h})). \tag{1}$$

Then, we conduct a cross entropy loss L_{node} between the prediction from Equation (1) and ground truth node types to optimize the graph encoder.

5.2 BART as Conversational Recommender

BART is a Transformer-based (Vaswani et al., 2017) sequence-to-sequence model, which can be seen as the generalizing BERT (bidirecitonal encoder) and GPT (autoregressive decoder). In the design of BART, the decoder performs cross-attention from each of its layers over the final hidden state of the encoder to be aware of input sequences. This operation seamlessly integrates the recommendation and dialogue modules into a unified conversational recommender.

BARCOR features four advantages over the graph-based recommender in the previous works: First, a unified framework inherently fuses the semantics between the encoder and the decoder and becomes less sensitive to the design of model architecture and hyper-parameters selections. In contrast, other works propose complex attentive interactions between modules, which is not robust from an actual production system perspective. That is, slight parameter changes would impact the performance. Moreover, BART is proved to be effective in various downstream tasks, such as neural machine translation and question answering. Secondly, BART takes users' utterances as inputs without further processing. Instead, in Zhou et al. (2020), the graph-based recommender demands manual annotations for movies and words in input texts to build a user preference, which is impractical under a realistic scenario. Thirdly, the learned knowledge from pretrained models provides rich sentence semantics. Finally, BART can perform an end-to-end training scheme for both the recommendation and generation tasks. Conversely, other works tend to design separate modules for two tasks and further sequentially optimize each module.

Bidirectional Recommender Given a conversation context X, BART encoder transforms X into $\mathbf{c}$, the hidden state of the final self-attentive layer. Then, $\mathbf{c}$ is viewed as a sentence representation of X and also a search key for retrieving recommendation candidates. To derive the probability over the candidates, we apply inner-product to compute the similarity between $\mathbf{c}$ and entity representations $\mathbf{H}$ from the graph encoder,

$$\mathbf{p}_{\text{rec}} = \text{Softmax}(\mathbf{c}\mathbf{H}^{\intercal}), \tag{2}$$

$$\mathbf{p}_{\text{rec-infer}} = \text{Softmax}(\mathbf{c}\mathbf{H}_I^{\intercal}), \tag{3}$$

where $\mathbf{p}_{\text{rec}} \in \mathbb{R}^{|\mathcal{E}|}$ denotes the recommendation prediction. To learn parameters in BARCOR, we employ a cross-entropy loss L_{rec} between the prediction from Equation (2) and the labels of ground truth entities. Note that the search space of recommended candidates is $\mathbf{H}$, which means both *movie items* and *descriptive entities* are likely to be retrieved.

Data Augmentation Since sentence-level semantics extracted from BART encoder is naturally inconsistent with entity-level semantics from the graph encoder, other than optimizing BARCOR by L_{rec}, we propose to (1) augment the training set with descriptive entities and (2) strategically initialize the graph encoder's embeddings to facilitate the fusion of heterogeneous semantics. First, during training, we construct data using the names of descriptive entities as the conversation context, such as "George Clooney," and the entities themselves as the recommended items. The data allows the representations of descriptive entities to be directly optimized by L_{rec} instead of optimized indirectly through their one-hop neighboring movie items. Besides, BARCOR becomes more aware of their names in conversation context and neighboring movie items. Secondly, we initialize entity embeddings $\{\mathbf{e}_i\}_{i=1}^{|\mathcal{E}|}$ with the sentence representations of their names transformed by the pretrained BART encoder. Thus, the initial semantic gap between two types of representations becomes closer, presumably easier to fuse. However, during the inference phase, the search space is reduced to the item set $\mathcal{I}$. The recommendation prediction is computed through Equation (3), where $\mathbf{H}_I$ is the matrix only consisting of movie item representations.

Auto-Regressive Response Generator We retain the original operations of BART decoder, which is conditioned on an input sequence and its sentence representation (i.e., the final hidden state of BART encoder) to generate a response autoregressively. Therefore, we follow Radford and Narasimhan (2018) to compute the generative probability and optimize the decoder through negative log-likelihood. During training, we mask the target responses of the augmented dataset to preserve authentic conversation flows.

End-to-End Training We optimize BARCOR by simultaneously performing the recommendation and generation tasks, compared to previous works demanding sequential optimization for two separated components. That is, we jointly minimize the objective as follow:

$$L = L_{\text{rec}} + \alpha L_{\text{gen}} + \beta L_{\text{node}},$$

where α and β are hyper-parameters determined by cross-validation.

6 Experiments

6.1 Experiment Setup

Baselines We compare BARCOR with the following baseline methods for the recommendation and response generation tasks on the processed ReDial dataset as discussed in Section 3.

- **KBRD** (Chen et al., 2019) employs DBpedia to enhance semantics of contextual items or entities for the construction of user preferences. The dialogue module is based on Transformer, where KG information is incorporated as word bias during generation.
- **KGSF** (Zhou et al., 2020) uses MIM (Viola and Wells, 1995) to fuse the information of entity-oriented and word-oriented KGs (i.e., DBpedia and ConceptNet). A user preference is constructed by fused representations of items and words. The dialogue module is based on Transformer, consisting of a standard encoder and a KG-enhanced decoder.

Automatic Evaluation For the recommendation task, we adopt *Recall@k* (R@k, k=1, 5, 10, 50), which suggests whether top-k recommended items contain the ground truth recommendations for evaluation. Since users may be frustrated by too many recommendations within a response, Recall@1,5 more faithfully present the recommendation performance. For the generation task, we follow Zhou et al. (2020) to use *Distinct n-gram (Dist-n, n=2, 3, 4)*, which measures the diversity of sentences. Since CRSs interact with humans through natural language, we introduce two metrics to capture the effectiveness of recommendations. *Item-F1* measures whether a CRS accurately provides recommendations compared to ground truth responses. *Average Item Number (AIN)* denotes the average number of recommended items within a sentence and presents the informativeness of generated responses.

Human Evaluation Aligning the CRS goal of providing successful recommendations, we invite 11 professional annotators to judge response quality. Given 40 multi-turn conversations from the testing set, the annotators evaluate the quality in terms of three aspects: (1) *Fluency*, (2) *Relevancy*, and (3) *Informativeness*, with each score ranging from 0 to 2.

6.2 Result Analysis

Table 3 summarizes the performance of different methods on the ReDial dataset, including human evaluation and automatic evaluation for the recommendation and response generation tasks.

Item Recommendation As we can see, KGSF outperforms KBRD because KGSF incorporates a word-oriented KG to enrich entity representations, highlighting the importance of words in context for the representation learning. With learned knowledge from pretrained models, BARCOR achieves 2.53% in R@1, 9.98% in R@5, 16.17% in R@10, and 34.95% in R@50 and outperforms KGSF by 79% and 30% in terms of R@1 and R@5 respectively. It demonstrates a tight fusion of semantics between sentences in context and entities in KG. Also, context and knowledge provide richer entity information, compared to the word-oriented KG adopted by KGSF.

	Model	Recommendation				Response Generation					Human Evaluation		
		R@1	**R@5**	**R@10**	**R@50**	**Dist-2**	**Dist-3**	**Dist-4**	**Item-F1**	**AIN**	**Fluen.**	**Relev.**	**Informat.**
(a)	KBRD	1.46	7.23	12.65	30.26	14.32	27.27	39.57	58.80	36.63	1.62	1.08	1.01
(b)	KGSF	1.41	7.66	13.47	32.17	19.49	35.36	49.19	62.61	41.00	1.56	0.98	0.66
(c)	**BARCOR**	**2.53**	**9.98**	**16.17**	**34.95**	**58.90**	**88.75**	**102.52**	**71.71**	**53.00**	**1.86**	**1.76**	**1.57**
(e)	(c) - Node Loss	2.32	9.01	15.61	34.3	41.12	61.15	73.60	71.08	45.22	-	-	-
(f)	(c) - Data Aug.	2.23	9.22	14.62	34.16	31.91	45.05	53.57	55.13	44.64	-	-	-
(g)	(c) - Node Init.	1.95	8.68	14.67	33.86	22.32	35.33	45.19	68.21	44.30	-	-	-
(h)	(c) - CORG	2.29	9.15	15.32	33.34	30.50	43.11	50.80	70.00	48.37	-	-	-

Table 3: Results on the recommendation and response generation tasks. In human evaluation, "Fluen.", "Relev", and "Informat" denote fluency, relevancy, and informativeness, respectively. The best results are in bold.

Response Generation In the automatic evaluation, the proposed BARCOR outperforms all baseline methods with a large margin in terms of Dist-n. Compared to KGSF, it improves Dist-2, Dist-3, and Dist-4 by +39.41%, +53.39%, and +53.33%, respectively, which demonstrates the proposed method effectively generates diverse sentences. Besides, BARCOR achieves 71.71% in Item-F1 and 53% in AIN. It suggests that BARCOR interprets user intentions to further precisely generate responses containing recommendations. In the human evaluation, BARCOR performs best among all baseline methods for the three metrics. We can note that BARCOR especially has higher scores of Relevancy and Informativeness, indicating generated responses are both accurately aligned with user intentions and rich in recommended items and related information. It verifies our interpretation of the scores of Item-F1 and AIN in the automatic evaluation. The above results prove the effectiveness of our method that fuses entity representations from the KG with sentence representations to generate fluent, relevant, and informative utterances. We also provide qualitative analysis in Appendix B.

Training Stability Figure 3 shows the performance curves of Recall@5 (R@5) and recommendation loss on the validation set for different methods. We select R@5 as the evaluation metric since it is neither too strict nor tolerable for accurate recommendations. It can be observed that BARCOR is more stably optimized and achieves a better performance than other competitive baseline methods. Within the first four epochs, both KBRD and KGSF quickly reach an optimal state where models gain the highest R@5 with the least recommendation loss. However, as training progresses, they begin to overfit the training data, leading to the decline in R@5 and the rise of the recommendation loss. The instability may be attributed to the insufficiency of semantics in conversation context and

Figure 3: Recommendation performance of BARCOR and the baselines on the validation set at different training epochs.

the number of trainable parameters. To construct a user representation, the baselines aggregate information of annotated entities, including movies and their associative properties, in conversation context. Although KGSF incorporates a word-oriented KG and a semantic fusion technique, the combinations of words and entities are still limited to the training set and the KGs. Therefore, some informative words or entities and their variants are lost if not presented in the corpus. In contrast, BARCOR directly encodes an entire context to build a user representation, ensuring every word is considered and increasing word semantic richness. Learned knowledge from pretrained models also prevents BARCOR from overly biasing on the training set. Moreover, we note that the number of trainable parameters of the BARCOR's recommendation module (39 million) is less than half of that of KGSF's (106 million) and KBRD's (91 million) recommenders. More details about models is presented in Table 5 in Appendix. Optimized fewer parameters with inputs of richer semantics, BARCOR consistently outperforms these baselines for all recommendation metrics. The results demon-

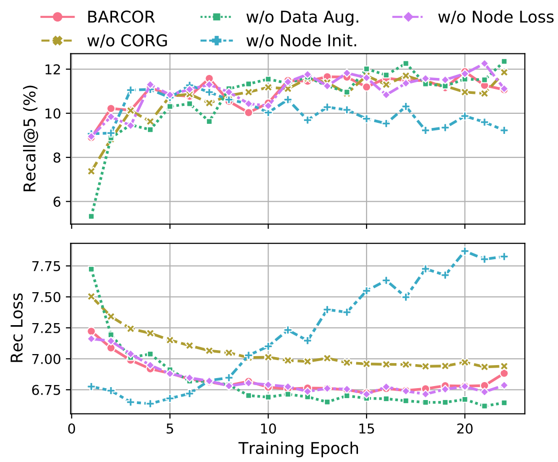

Figure 4: Ablation study: Recommendation performance on the validation set at different training epochs.

strate the effectiveness and optimization stability of the proposed unified framework for modeling CRS.

6.3 Ablation Study

To understand the contribution of each component on the recommendation and generation tasks, we construct a ablation study for four variants in BARCOR: (1) BARCOR (w/o Node Loss): removing cross entropy loss of the node classification task presented in Section 5.1, (2) BARCOR (w/o Data Aug.): removing the training set augmentation mentioned in Section 5.2, (3) BARCOR (w/o Node Init.): replacing node embeddings from the pretrained BART encoder by randomly initialized weights mentioned in Section 5.2, and (4) BARCOR (w/o CORG): excluding CORG by removing relations among nodes.

Since the recommendation and dialogue modules share the same sentence representation of context, techniques designed for representation enrichment are mutually beneficial for both tasks. As shown in Table 3 (row(e-h)), all techniques are helpful to improve the final performance in terms of all metrics. Besides, node embeddings initialization of the graph encoder and the proposed CORG are seemed to be more critical. First, we observe that R@1, R@5, and Dist-n decrease when the node embeddings are randomly initialized. Also, the validation performance curves in Figure 4 reveal the issue of overfitting, as shown in Section 6.2. We attribute this to the increased optimization difficulty brought by the incorporation of the graph encoder. The number of its trainable parameters is 27 million, accounting for 68% of the total trainable parameters in the recommendation module. Randomly initialized embeddings easily fit the seen data but difficultly fuse with sentence semantics from the BARCOR's encoder. The results reinforce our claim discussed in Section 6.2. Although random initialization leads to the decline in performance, BARCOR (w/o Node Init.) still outperforms the strong baselines for all evaluation metrics. Second, as shown in row(h), BARCOR (w/o CORG) surprisingly achieves competitive results with BARCOR in R@1, R@5, and R@10 and outperforms KGSF using two KGs. Namely, BARCOR (w/o CORG) merely leverages relations of entities and words in the dialogue history to recommend more accurately than the KG-enhanced strong baselines. It implies that implicit relations of entities within context have yet been exploited to the fullest.

In conclusion, the sentence-level semantics derived from BARCOR's encoder provide richer information than the entity representations encoded by the R-GCN, and is sufficient for accurate recommendations. Besides, a trade-off between KG-based information enrichment and optimization difficulty for a graph encoder needs careful consideration. In our work, we propose incorporating supervision signal from the node classification task, training set augmentation, and node embeddings initialized by the pretrained BART to reduce the difficulty. We hope these results inspire future research.

7 Conclusion

In this paper, we proposed a novel unified framework for the conversational recommendation, BARCOR. BARCOR jointly tackles the recommendation and generation tasks with the shared sentence representation of conversation history. It serves as a search key for item retrieval and provides rich fused semantic of sentences and entities for the decoder to generate responses. Moreover, we enrich the information of entities by constructing a high-quality KG, namely CORG, and incorporating a graph encoder exploiting structural knowledge. The experiments results demonstrate that BARCOR achieves superior performance on recommendation accuracy and response quality than all competitive baselines and generates informative responses with fluency and relevancy.

References

Anders H. Brams, Anders L. Jakobsen, Theis E. Jendal, Matteo Lissandrini, Peter Dolog, and Katja Hose. 2020. Mindreader. *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*.

Qibin Chen, Junyang Lin, Yichang Zhang, Ming Ding, Yukuo Cen, Hongxia Yang, and Jie Tang. 2019. Towards knowledge-based recommender dialog system. *arXiv preprint arXiv:1908.05391*.

Konstantina Christakopoulou, Alex Beutel, Rui Li, Sagar Jain, and Ed H Chi. 2018. Q&r: A two-stage approach toward interactive recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 139–148.

Matthias Fey and Jan Eric Lenssen. 2019. Fast graph representation learning with pytorch geometric.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Raymond Li, Samira Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards deep conversational recommendations.

Shijun Li, Wenqiang Lei, Qingyun Wu, Xiangnan He, Peng Jiang, and Tat-Seng Chua. 2020. Seamlessly unifying attributes and items: Conversational recommendation for cold-start users. *arXiv preprint arXiv:2005.12979*.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29.

Yu Lu, Junwei Bao, Yan Song, Zichen Ma, Shuguang Cui, Youzheng Wu, and Xiaodong He. 2021. RevCore: Review-augmented conversational recommendation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1161–1173, Online. Association for Computational Linguistics.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

Rajdeep Sarkar, Koustava Goswami, Mihael Arcan, and John McCrae. 2020. "suggest me a movie for tonight": Leveraging knowledge graphs for conversational recommendation.

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2017. Modeling relational data with graph convolutional networks.

Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 553–562.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

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

Zhu Sun, Jie Yang, Jie Zhang, Alessandro Bozzon, Long-Kai Huang, and Chi Xu. 2018. Recurrent knowledge graph embedding for effective recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 297–305.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

P. Viola and W.M. Wells. 1995. Alignment by maximization of mutual information. In *Proceedings of IEEE International Conference on Computer Vision*, pages 16–23.

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*, pages 417–426.

Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 950–958.

Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 353–362.

Tong Zhang, Yong Liu, Peixiang Zhong, Chen Zhang, Hao Wang, and Chunyan Miao. 2021. Kecrs: Towards knowledge-enriched conversational recommendation system.

Xiaoying Zhang, Hong Xie, Hang Li, and John C.S. Lui. 2020. Conversational contextual bandit: Algorithm and application. *Proceedings of The Web Conference 2020*.

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*, pages 1006–1014.

Measure	Value
# Node	23,164
# Movie	6,924
# Genre	313
# Cast Member	11,017
# Director	3,587
# Production Company	1,323
# Edge	87,212
# Movie-Genre	19,292
# Movie-Cast Member	53,109
# Movie-Director	7,155
# Movie-Production Company	7,407
# Genre-Genre	249

Table 4: Graph statistics of the constructed CORG.

Model	Rec.	Gen.	# Total
KBRD	85.9 %	14.1 %	105,601,166
KGSF	81.6 %	18.4 %	129,899,342
BARCOR	53.8 %	46.2 %	72,593,777

Table 5: Model statistics. "Rec." and "Gen." represent the recommendation and response generation modules, respectively. These two columns present the ratio of the trainable parameters in the two modules to the total number of trainable parameters, which is presented in the column "# Total." The statistics of KBRD[5] and KGSF[6] are derived from the publicly released implementations of their authors on GitHub.

A Implementation Details

In all the experiments, we use mini-batch AdamW with learning rate 3×10^{-5} as the optimizer and each batch of 64 examples on a single Nvidia Tesla V100. The whole training takes 22 epochs without early stop. The entire implementation was based on PyTorch, PyTorch Geometric (Fey and Lenssen, 2019), and HuggingFace transformers[4] package. We finetune the 11-th attention layer of BART encoder and the 10-th and 11-th attention layers of BART decoder for the CRS task. The detailed number of trainable parameters are listed in Table 5.

B Qualitative Analysis

In the section, we present several conversations generated by different models in Table 6. Compared to other baseline methods, BARCOR can understand the user intention to provide a relevant recommendation and generate informative responses related to the recommended item. In example (a), when the seeker asks for romantic comedy and mentions "*How to Lose a Guy in 10 Days (2003)*", BARCOR recommends another romantic comedy "*50 First Dates (2004)*". Besides, it also expresses the attitude toward the recommended item and makes the response more informative by saying that "*is a good romantic comedy with Adam Sandler*." In example (b), BARCOR grasps the idea of great old movies and recommends "*Gone with the Wind*

(a)	
Seeker:	Hello, I really need a good movie for tonight. I have a date.
Recommender:	Ok.
Seeker:	Maybe a romantic comedy like **How to Lose a Guy in 10 Days (2003)**.
KBRD:	I haven't seen that one yet.
KGSF:	**Love Actually** is a good one.
BARCOR:	**50 First Dates (2004)** is a good romantic comedy with Adam Sandler.
Human:	How about **10 Things I Hate About You (1999)**?
(b)	
Seeker:	Hello! How are you?
Recommender:	Hey there. I'm doing well. Can I recommend a movie for you?
Seeker:	I'm looking for a great old movie. Any suggestions?
KBRD:	Sure! Have you seen **Black Panther (2018)**?
KGSF:	I am doing great.
BARCOR:	Have you seen **Gone with the Wind (1939)**?
Human:	Hmmmm, such as **Breakfast at Tiffany's (1961)**? I loved that movie.
(c)	
Recommender:	Looking for a good movie?
Seeker:	I have a movie night tonight. My firends are coming. I need good mystery movies.
Recommender:	Well, **Wind River (2017)** is a good mystery.
Seeker:	I have seen that one.
KBRD:	It is really good.
KGSF:	**Shutter Island (2010)** is a good one.
BARCOR:	**Gone Girl (2014)** is another good one. If you're looking for something a little more graphic, it is one of my favorite movies.
Human:	How about **Memento (2000)**?

Table 6: Examples of generated responses from different models. Movie names are in bold.

[4] https://huggingface.co/transformers/

(1939)", an epic historical romance film. Conversely, KBRD simply recommends a well-known modern movie, which fails to meet the user demand. In example (c), when asked a mystery movie like "*Wind River (2017)*", the human recommender and KGSF merely give recommendations without personal insight. However, BARCOR not only recommends another mystery movie, "*Gone Girl (2014)*", but explains the motivation behind the recommendation by saying that "*If you're looking for something a little more graphic, it is one of my favorite movies*."