Improving Conversational Recommendation Systems' Quality with Context-Aware Item Meta-Information

Anonymous ACL submission

Abstract

A key challenge of Conversational Recommendation Systems (CRS) is to integrate the recommendation function and the dialog generation function smoothly. Previous works employ graph neural networks with external knowledge graphs (KG) to model individual recommendation items and integrate KGs with language models through attention mechanisms for response generation. Although previous approaches prove effective, there is still room for improvement. For example, KG-based approaches only rely on entity relations and bag-of-words to recommend items and neglect the information in the conversational context. We propose to improve the usage of dialog context for both recommendation and response generation using an encoding architecture along with the self-attention mechanism of transformers. In this paper, we propose a simple yet effective architecture comprising a pre-trained language model (PLM) and an item metadata encoder to integrate the recommendation and the dialog generation better. The proposed item encoder learns to map item metadata to embeddings reflecting the rich information of the item, which can be matched with dialog context. The PLM then consumes the context-aware item embeddings and dialog context to generate high-quality recommendations and responses. Experimental results on the benchmark dataset REDIAL show that our model obtains state-of-the-art results on both recommendation and response generation tasks[1].

1 Introduction

An automated conversational recommendation system (CRS) (Li et al., 2018; Zhou et al., 2020) is intended to interact with users and provide accurate product recommendations (e.g., movies, songs, and consumables). It has been a focal point of research lately due to its potential applications in the e-commerce industry. Traditional recommendation systems collect user preferences from implicit feedback such as click-through-rate (Zhou et al., 2018) or purchase history and apply collaborative filtering (Su and Khoshgoftaar, 2009; Shi et al., 2014) or deep learning models (Covington et al., 2016; He et al., 2017) to construct latent spaces for user preferences. Unlike traditional recommendation systems, CRSs directly extract user preferences from live dialog history to precisely address the users' needs.

Although some progress has been made in this area, there is still room for improvement. First, previous CRSs (Chen et al., 2019; Zhou et al., 2020; Li et al., 2021) track entities mentioned in the dialog context, and then search related items in knowledge graphs to recommend to users. However, these systems require a named-entity recognition (NER) module to extract mentioned entities from the dialog context. Thus we need to collect additional domain-specific data to train the NER module. In practice, such NER modules have deficient performance, leading to a bad accuracy of CRS. Second, existing CRSs built upon graph neural networks (Kipf and Welling, 2017; Schlichtkrull et al., 2018) cannot quickly scale up or respond to rapid changes of the underlining entities. In e-commerce, items for recommendation change frequently due to constant updates of merchants and products. Existing approaches require either re-training the entire system when the structure of knowledge graph changes (Dettmers et al., 2018) or adding complex architectures on top to be adaptive (Wu et al., 2019). A more flexible architecture can help the system react to rapid changes and adapt itself to new items.

Moreover, meta-information about the items can be leveraged. Similar information can be found in both dialog context and item meta-information. For example, in a movie recommendation setting, words like "crime, gangsters, etc." are likely to exist in the dialog context when a user is searching

[1] Code is available online https://github.com/by2299/MESE

for crime movies. In the synopsis of a crime movie, such keywords are likely to exist as well. Ideally, an alignment between the semantics of dialog context and the item meta-information can be used to improve system's performance.

Driven by the motivations above, we present a **M**etadata **E**nhanced learning approach via **S**emantic **E**xtraction from dialog context i.e. **MESE**. The major components of MESE contain a pre-trained language model (PLM) and an item encoder architecture. The item encoder takes item metadata as input and outputs a vector embedding. By jointly training the encoder and the PLM, the entire system can extract co-occurring information between dialog context and item metadata, and item encoders can systematically construct representations reflecting this alignment. Item embeddings are then consumed with dialog context by the self-attention mechanism of the PLM. This mechanism smoothly integrates dialog context and item information well into the recommendation and response generation tasks.

The key contributions of this paper are summarized as follows: This paper presents MESE, a novel CRS framework that considers both item metadata and dialog context for recommendations. Our model employs a simple yet effective item metadata encoder that learns to represent rich item information during training. Such encoder can adapt to database changes quickly and is independent of task-specific architectures. Extensive experiments on standard dataset REDIAL demonstrate that MESE outperforms previous state-of-the-art methods on both response generation and recommendation with a large margin.

2 Related Work

The current CRS paradigm contains two major modules: a recommendation module that suggests items based on conversational context and a response generation module that generate responses based on dialog history and the recommended items. Integrating these two modules to perform well on both tasks has been a major challenge. Chen et al. (2019) leverage external knowledge and employees graph neural networks as the backbone to model entities and entity relations in the knowledge graph (KG) to enhance performance. Zhou et al. (2020) introduce a word-level KG (Speer et al., 2017) to the system with semantic fusion (Sun et al., 2019a) to enhance the semantic representations of words and items. Since item information and dialog context are processed separately in the above approaches, they loss integrated sentence-level information. We propose to condition recommendation on integrated contextual information of both dialog context and mentioned entity information. More recent works adopt pre-trained language models (PLM) (Vaswani et al., 2017; Radford et al., 2019; Zhang et al., 2020) and template-based methods to facilitate response generation. Liang et al. (2021) generate a response template containing a mixture of contextual words and slot locations to incorporate recommended items better. Wang et al. (2021) expand the vocabulary list of the PLM to include items to unify the process of item recommendation with response generation. We propose to enhance our PLM with an item metadata encoder to extract context-aware representations by jointly training on both recommendation and response generation tasks. We also generate response templates with slot locations to better incorporate recommended items into responses.

Our work is also inspired by studies from other areas. Recent works have shown that cross-modality training across vision and language tasks can lead to outstanding results in building multimodal representations (Tan and Bansal, 2019; Lu et al., 2019). In (Tan and Bansal, 2019), a large-scale transformer-based model is adapted with cross-modal encoders to connect visual and linguistic semantics and pre-trained on vision-language pairs to learn cross-modality relationships. Prompt tuning (Li and Liang, 2021; Gao et al., 2021) methods show that PLMs are capable of integrating different sources of information into the same embedding space. In terms of using PLM as a recommendation system, Sun et al. (2019b) train a bidirectional self-attention model to predict masked items and achieve remarkable results. Inspired by the above studies, we propose to use an encoder module to map item meta-information to an embedding space. By jointly training on dialog context and encoded item representations, the system can align these two information streams by fusing the semantic spaces.

3 Approach

In this section, we present our framework MESE that integrates item metadata with dialog context. We first introduce how to encode item metadata and how to blend item information into dialog context. We then illustrate how the recommendation module and the response generation module are built. Finally, we describe the training objectives and the testing process.

3.1 Encoding Item Metadata

We propose to use an item encoder to directly map the metadata of each item to an embedding. In the movie recommendation setting, description on title, genre, actors, directors, and plot are collected as metadata and concatenated with a "[SEP]" token for each movie. This concatenated information is the input to the item encoder which produces a vector representation for each item. The item encoder consists of a DistilBERT (Sanh et al., 2019) model that maps the input sequence to a sequence of vector embeddings, a pooling layer that condenses the sequence embeddings to a single vector embedding, and a feed-forward layer to produce the output embedding with a certain dimension. A visualization of this module is shown in Figure 1.

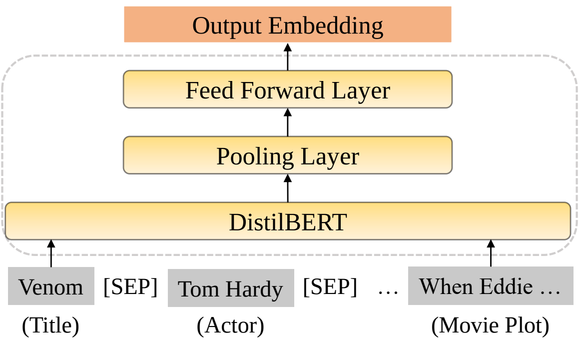

Figure 1: Item Encoder takes in the metadata of an item and outputs an embedding of the item

Next, we discuss how to incorporate items into dialog context with the encoded embeddings and the PLM (Radford et al., 2019). Previous studies have shown that KG-based frameworks cannot always integrate recommended items into generated replies (Wang et al., 2021). To solve this issue, we introduce a special placeholder token "[PH]" to the vocabulary list of the PLM. Every occurrence of item name in the corpus is replaced with this "[PH]" token. This modified dialog sequence is then mapped to a sequence of word token embeddings (WTE) by the vocabulary embedding matrix of the PLM. To include item information into the context, an instance of the item encoder is used to encode item metadata into token embeddings. The item encoder takes in item metadata and outputs an item token embedding (ITE) with the same dimensionality as a WTE of the PLM. The ITE is then concatenated with the WTEs constructed from the dialog context to be consumed by the PLM. An example is shown in 2.

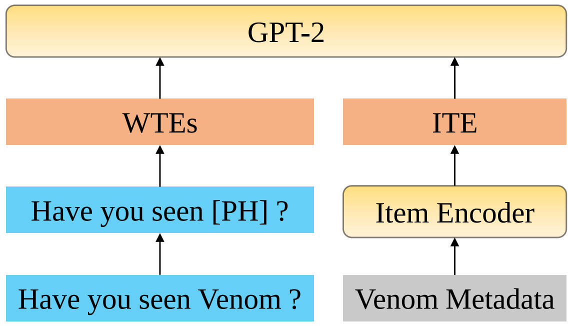

Figure 2: Dialog context is represented as a concatenation of WTEs and ITEs to be consumed by the PLM.

3.2 Recommendation Module

Similar to (Covington et al., 2016), we pose recommendation as a two-phase process: candidate selection and candidate ranking. During candidate selection, the entire item database is traversed and narrowed down to a few hundred candidates based on a calculated similarity score between the dialog context and the item metadata. During candidate ranking, similarity scores between the dialog context and the generated candidates are recomputed with finer granularity by the self-attention mechanism of the PLM.

3.2.1 Candidate Selection

In this section, we describe the training objective of candidate selection. We add a special token "[REC]" to the vocabulary embedding matrix of PLM. This token is used to indicate the start of the recommendation process and to summarize dialog context. At the end of each turn, a token embedding sequence is created following Figure 2 in the format of an interleaving of word token embeddings (WTE) and item token embeddings (ITE) to represent all previous dialog context. When recommendation is labeled in a conversation turn in the training dataset, the WTE of "[REC]" is appended to the previous token embedding sequence to form a new sequence D. Next, the PLM takes in D and produces an output embedding sequence. We denote the last vector of this output embedding sequence as D_R which corresponds to the appended special token "[REC]". D_R summarizes dialog context and can be used to retrieve candidate items.

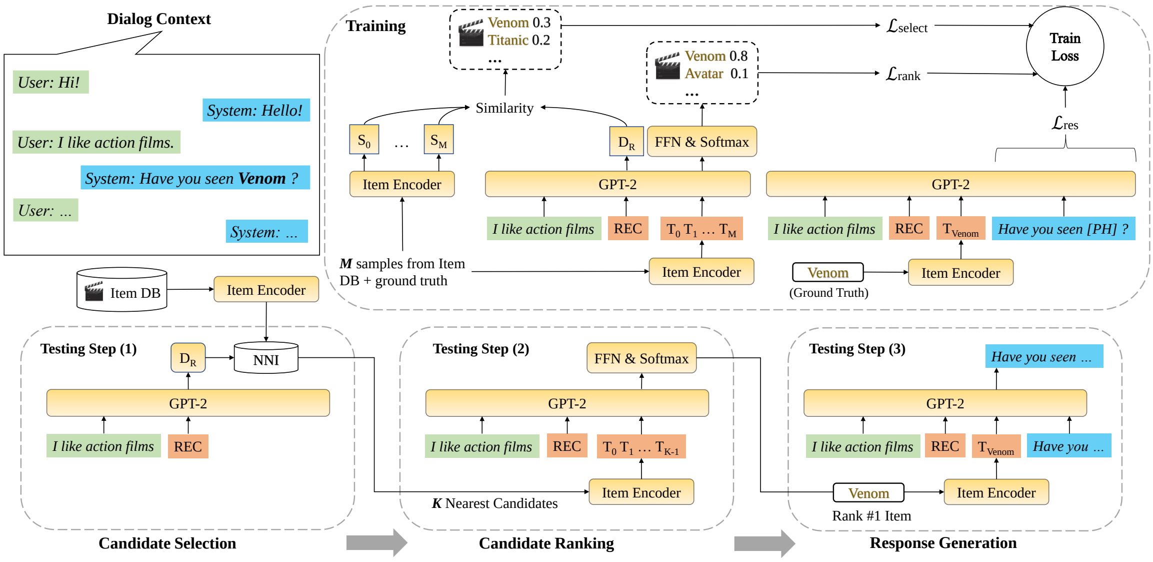

Figure 3: Overview of MESE. During training, M items are sampled from the database to compute the joint loss $\mathcal{L}_{\text{select}}$ and $\mathcal{L}_{\text{rank}}$, which are then combined with the response generation loss $\mathcal{L}_{\text{res}}$ and jointly optimized. During testing, the entire metadata DB is stored as a nearest neighbor index (NNI). First, dialog context is condensed into a vector D_R. An approximate nearest neighbor search is performed on D_R to get candidate items, which is then passed to the ITE Encoder to compute their ranking scores and the the highest-ranked candidate is used as a prompt to generate responses. We only present the case when there's only one ground truth recommendation in the utterance. However, it's easy to extend the above approach to multiple recommendations.

We randomly sample M items and their metadata from the database as negative examples and combine them with the ground truth item labeled in the dataset to get the training samples. Another instance of the item encoder, is used to create candidate token embeddings for each item in the training samples. The item Encoder takes in the metadata of samples items and outputs a set of candidate token embeddings $C = (c_0, c_1, ..., c_M)$, each with the same dimensionality as D_R. The recommendation task at this phase is posed as a multi-class classification problem of predicting the ground truth item over the negative samples (Covington et al., 2016). The probability of each candidate item is defined in (1) and optimized by a cross-entropy loss function, denoted as $\mathcal{L}_{\text{select}}$:

$$P(i) = \frac{e^{c_i \cdot D_R}}{\sum_{n=0}^{M} e^{c_n \cdot D_R}} \tag{1}$$

Note that the purpose of this learning objective is to let the model learn how to construct the D_R representation instead of learning the probabilities of candidate items. The D_R representation is later used in an approximate nearest neighbor search (Liu et al., 2004) to select candidates from the entire database in testing 3.5.

3.2.2 Candidate Ranking

In this section, we describe the training objective of candidate ranking. The goal of candidate ranking is to further perform more fine-grained scoring on the similarities between generated candidates and dialog context so that the final rankings of items can better reflect users' preferences. We propose to use the PLM and its self-attention to compute ranking scores.

During training, the same context token embedding sequence D and the same training sample with M negative examples are used. The ITE encoder from section 3.1 is used to map the metadata of the sample to an ITE set $T = (t_0, t_1, ..., t_M)$, where the subscript of each t_i corresponds to their index in the database. A concatenation of context sequence D and T are created and consumed by the same PLM used above and the output embeddings are computed. The order of candidate items should not make a difference in the values of the outputs. Therefore, we add the same positional encoding to each ITE in T and remove the attention masks among the ITEs. The output embeddings of PLM that correspond to the ITEs in T are then passed to a feed-forward layer to reduce each vector from a higher dimension to a single number with dimensionality equals 1. This set of numbers is denoted by Q = $(q_0, q_1, ..., q_M)$ where the index of each number corresponds to their index in T. The final ranking score of each candidate item is defined in (2) and optimized by a cross-entropy loss function, denoted as $\mathcal{L}_{\text{rank}}$:

$$R(i) = \frac{e^{q_i}}{\sum_{n=0}^{M} e^{q_n}} \tag{2}$$

3.3 Response Generation Module

In this section, we describe how to train the model to generate responses based on the recommended items' metadata. The same token embedding sequence D is used as context and current system utterance $U = (w_0, w_1, ..., w_n)$ is used as targets where each w_i represents a WTE. We only optimize the PLM to reconstruct system utterances.

If the current utterance contains recommendations, we create ITEs by passing metadata of the recommended items through the item Encoder used in 2 and append the ITEs to context token embedding sequence D to obtain D'. If the current utterance doesn't contain recommendations, D' is the same as D. The PLM is trained to reconstruct the ground truth U based on D'. The probability of generated response is formulated as:

$$P(U|D') = \prod_{i=1}^{n} P(w_i|w_{i-1}, ..., w_0, D') \tag{3}$$

The loss function is set to be:

$$L_{res} = -\frac{1}{N}\sum_{i=1}^{N} log(P(U_i|D')) \tag{4}$$

Where N is the total number of system utterances in one dialog.

3.4 Joint Training

Finally, we use the following combined loss to jointly train both the encoders and the PLM:

$$Loss = a \cdot \mathcal{L}_{\text{select}} + b \cdot \mathcal{L}_{\text{rank}} + c \cdot \mathcal{L}_{\text{res}} \tag{5}$$

Where a, b and c are the weights of language training and recommendation training objectives. During training, all weight parameters of the two item encoders, the PLM and relevant feed-forward layers participate in back-propagation. An overview of training is shown in Figure 3

3.5 Testing

During testing, a candidate embedding set over the entire item database is built by running metadata through the item encoder used in section 3.2.1 and stored with a nearest neighbor index (NNI) (Muja and Lowe, 2014). During response generation, when a "[REC]" token is generated, candidate selection 3.2.1 is activated. An approximate nearest neighbor search is conducted over the NNI and K closest candidates are selected based on their similarities from the D_R vector[2]. Candidate ranking is then activated and the PLM and the item encoder from 2 are used to generate a score for each candidate. When ranking finishes, the ITE that receives the highest ranking score is appended to the dialog context D and response generation continues until the end-of-sentence token is generated. After generation is completed, we replace the occurrence of the placeholder token "[PH]" with the title of the recommended item to form the final response. Note that when the turn involves no recommendation, our PLM simply generates a clarification question or a chitchat response with no placeholder tokens. An overview of testing is shown in Figure 3

4 Experiments

In this section, we discuss the datasets used, experimental setup, experimental results on both recommendation and language metrics, and report analysis results with ablation studies.

4.1 Datasets

We evaluated our model on two datasets: ReDial dataset (Li et al., 2018) for comparison with previous models and INSPIRED dataset (Hayati et al., 2020) for ablation studies. Both datasets were collected on Amazon Mechanical Turk (AMT) platform where workers made conversations related to movie seeking and recommending following a set of extensive instructions. The statistics of both datasets are shown in Table 1

Dataset	dialogs	utterances	avg turns
ReDial	10006	182150	18.2
INSPIRED	1001	35811	10.73

Table 1: Statistics of Datasets

[2]Multi-Source Selection in Appendix A

4.2 Experimental Setup

4.2.1 baselines

The baseline models for evaluation on the ReDial dataset is described below:

ReDial (Li et al., 2018): A dialogue generation model using HRED (Sordoni et al., 2015) as backbone for dialog module

KBRD (Chen et al., 2019): The dialog generation module based on the Transformer architecture (Vaswani et al., 2017). It exploits external knowledge to perform recommendations and language generation.

KGSF (Zhou et al., 2020): Concept-net is used alongside knowledge graph to perform semantic-aware recommendations.

CR-Walker (Ma et al., 2021): performs tree-structured reasoning on a knowledge graph and guides language generation with dialog acts

CRFR (Zhou et al., 2021): conversational context-based reinforcement learning model with multi-hop reasoning on KGs.

NTRD (Liang et al., 2021): an encoder-decoder model is used to generate a response template with slot locations to be filled in with recommended items using a sufficient attention mechanism.

RID (Wang et al., 2021): pre-trained language model and knowledge graph are used to improve CRS performance.

4.2.2 Implementation Details

We employed GPT-2 model (Radford et al., 2019) as the backbone of MESE for dialog generation, which contains 12 layers, 768 hidden units, 12 heads, with 117M parameters. We recruited 2 item encoders (Sanh et al., 2019) to encoder items in candidate generation 3.2.1 and candidate ranking 3.2.2, respectively, each has a distil-bert model with 6 layers, 768 hidden units, 12 heads, with 66M parameters. We used the AdamW optimizer (Loshchilov and Hutter, 2019) with epsilon set to $1e^{-6}$, learning rate set to $3e^{-5}$. The model was trained for 8 epochs on ReDial dataset, and the first epoch was dedicated to warm up with a linear scheduler. We set the sample size M during candidate generation and candidate ranking to be 150. We set a=0.8, b = 1.0 and c = 0.28 as coefficients for 3 loss functions respectively. We chose K = 500 for the number of candidates during testing.

4.2.3 Evaluation Metrics

We performed two evaluations, recommendation evaluation and dialog evaluation, for the model. For recommendation evaluation, we used Recall@X (R@X), which shows whether the top X items recommended by the system include the ground truth item suggested by human recommenders. In particular, we chose R@1, R@10 and R@50 following previous works (Chen et al., 2019; Zhou et al., 2020). We also defined recall accuracy of MESE to be the percentage of ground truth items that appear among the 500 generated candidates in the candidate generation phase 3.2.1 and ranking accuracy to be the percentage of items that appear in the top k (k=1, 10, 50) position of the sorted candidates in the candidate ranking phase 3.2.2. The product of the recall and ranking accuracy is the final recommendation accuracy of MESE. We also adopted end-to-end response evaluation following (Wang et al., 2021). We computed response recall (ReR) as whether the final response contains the target items recommended by human annotators. For dialog evaluation, we adopted perplexity, distinct n-grams (Li et al., 2016), and BLEU score (Papineni et al., 2002) for automatic evaluations. Human evaluation (on a random sampling of 100 dialogs from the test set) is also conducted on dialog evaluation in comparison with **KGSF**. We invite three annotators to score the generated samples in two aspects, Fluency and Informativeness. The annotator is asked to select a better response based on the given context. Ties are allowed if two responses have similar qualities. The score is the percentage of the model's response being selected. The final performance is calculated using the average scores of the three annotators.

5 Experimental Results

5.1 Evaluation Results

We first report recall, ranking, and final accuracy on REDIAL dataset of MESE in table 3. From the results, it can be seen that candidate ranking has remarkable performance gains in scoring the items. It demonstrates that PLMs have great potential in making recommendations. One possible reason behind this is that the PLM and its self-attention mechanism is effective in learning the similarities and discrepancies between item semantics and dialog semantics.

Table 2 compares different models on REDIAL dataset. The superiority of MESE persists across recommendation and language generation. On all recommendation metrics, including R@1, R@10, and R@50, MESE outperforms the state-of-the-art

Model	Recommendation metrics				Language generation metrics					
	R@1	R@10	R@50	ReR	PPL	Dist2	Dist3	Dist4	Bleu2	Bleu4
ReDial	2.4	14.0	32.0	0.7	28.1	0.225	0.236	0.228	0.178	0.074
KBRD	3.1	15.0	33.6	0.8	17.9	0.263	0.368	0.423	0.185	0.074
KGSF	3.9	18.3	37.8	0.9	5.6	0.289	0.434	0.519	0.164	0.074
CR-Walker	4.0	18.7	37.6	-	-	-	-	-	-	-
CRFR	4.0	20.2	39.9	-	-	-	-	-	-	-
RID	-	-	-	3.1	54.1	0.518	0.624	0.598	0.204	0.110
NTRD	-	-	-	1.8	**4.4**	0.578	0.820	1.005	-	-
MESE	**5.6**	**25.6**	**45.5**	**6.4**	12.9	**0.822**	**1.152**	**1.313**	**0.246**	**0.143**

Table 2: Results and comparison with the literature on REDIAL.

top k	Ranking Acc	Recall Acc	Final Acc
@1	7.2	0.778	**5.6**
@10	33.0	0.778	**25.6**
@50	58.5	0.778	**45.5**

Table 3: Recall, Ranking and Final Accuracy of MESE.

models by a large margin. We argue in 5.2 that this significant gain of performance is due to the effectiveness of the item encoder. MESE also performs well on the ReR score, which indicates that the filling placeholder tokens can help integrate recommended items into responses. For language generation, MESE also achieves significantly better performance than all other models on distinct ngrams and bleu scores with the exception that the PPL is worse than those of KGSF and NTRD. This indicates that MESE can generate more diverse responses while sticking to the topic.

Model	Fluency	Informativeness
KGSF	24%	19%
MESE	**38%**	**31%**

Table 4: Human Evaluation of Response Generation

Table 4 presents the results of human evaluation. Our proposed model MESE outperforms KGSF by a large margin on both fluency and informativeness. Responses of MESE have a 50% more chance of being chosen as the better answer than responses of KGSF. By using the encoded item embeddings and joint training, MESE can better integrate its pretrained weights with the injected item information. Therefore, it generates more fluent responses that contain richer information about the items.

5.2 Ablation Studies and Analysis

In this section, we first analyze the reason behind the performance gain of our recommendation module by analyzing the embeddings learned by the item encoder.

How much does metadata help recommendation? We argue that our training objectives on recommendation enable the item encoder to selectively extract useful features pertinent to the recommendation task from item metadata and construct item representations that resonate with instructional semantic properties in the dialog histories. For example, in REDIAL dataset, movie genre information is the most frequently mentioned property in dialog histories and human recommenders often make recommendation decisions based on this property. Although other properties like actors also help with recommendations, they do not appear in the corpus as often as genres or movie plots. We designed the following experiments to test our hypothesis. First, we train MESE with movie genre and plot information removed from the metadata, which we refer to as MESE w/o content, and compare its recommendation performance with MESE in Table 5.

Model	R@1	R@10	R@50
MESE w/o content	3.9	19.5	37.9
MESE	**5.6**	**25.6**	**45.5**

Table 5: Comparison Results of MESE and MESE w/o content.

As we can see from the table, there is a significant performance decrease after we remove genre and plot information, which indicates that MESE depends on the item information to make high-quality recommendations. We also point out that movie titles contain weak genre information but are not able to provide adequate features for the item encoder to extract from.

How does the item encoder help recommendation? We claim that the item encoder can construct embeddings in a systematic way that aligns matching information between its input and dialog context. We designed an experiment to prove the point. Specifically, we select all movie items with only one genre as our candidates, resulting in a subset of ~700 movies. We then select 2 item encoders (section 3.2.2) from MESE, MESE w/o content, and the item encoder before training (MESE raw), respectively, and obtain 3 sets of item embeddings of the selected movie subset. On each set of embeddings, we run a K-means clustering algorithm with K being set to be 3, 4, and 5, respectively. For each cluster obtained, we calculated the proportion of the majority genre among all item candidates. This process is repeated 20 times and the average accuracy is reported in Table 6. Genre information appears most frequently in dialog context and most recommendations are made based on genre attributes. Our item encoder, after joint training, should construct item embeddings that reflect genre information. Hence, the embeddings should be more clustered in terms of genre.

Model	K=3	K=4	K=5
MESE raw	0.492	0.514	0.574
MESE w/o content	0.555	0.589	0.606
MESE	**0.695**	**0.725**	**0.738**

Table 6: Item Encoders Clustering Accuracy

As we can see from the table, without training, MESE raw, being the least sensitive to genre information, achieves the lowest accuracy scores on all clusters. MESE w/o content, although deprived of genre and plot, still has slightly higher accuracy than MESE raw due to its exposure to REDIAL conversations. MESE is most sensitive to genre information. This is an indication that by aligning matching information in both dialog context and item metadata, our item encoder is able to generate meaningful representations, which can facilitate the PLM to produce better rankings through its self-attention mechanism.

What if we remove mentioned entities from dialog context? Mentioned entities are crucial to previous approaches (Chen et al., 2019; Zhou et al., 2020) in terms of recommendations. We train MESE with mentioned entities removed from dialog history and compare its performance with MESE on REDIAL dataset and INSPIRED dataset in table 7.

Dataset	Model	R@1	R@10	R@50
REDIAL	MESE w/o item	3.4	18.1	38.7
	MESE	**5.6**	**25.6**	**45.5**
INSPIRED	MESE w/o item	4.3	11.9	26.7
	MESE	**4.8**	**13.5**	**30.1**

Table 7: Results of MESE and MESE w/o on REDIAL and INSPIRED.

We can see removing the entities led to an average of 26.3% performance drop on REDIAL and an average of 11.2% performance drop on INSPIRED. The recommendation performance on REDIAL is more impacted by the removal of entities because the conversations in REDIAL are rich with entities and weak in semantic information, whereas INSPIRED is more sparse on entities but contains richer dialog information. In REDIAL, there is 1 mentioned movies among every 21.85 word tokens. The sentence level distinct 1-grams and 3-grams are 0.15 and 2.81. In contract, there is 1 mentioned movies among every 63.54 word tokens in INSPIRED. Its sentence level distinct 1-grams and 3-grams are 0.59 and 6.84. This proves that our model can efficiently infer user interests from texts to make high-quality recommendations without explicitly using mentioned entities. This property could be useful in an e-commerce setting where users tend to convey their requirements more with texts than entities. It could also be useful in a cold start scenario where we don't have many entities in the context.

6 Conclusion and Future Work

In this paper, we introduced MESE, a novel CRS framework. By utilizing item encoders to construct embeddings from metadata, MESE can provide high-quality recommendations that align with the dialog history. Our approach yields better performance than existing state-of-the-art models. Ablation studies explain the reason behind this performance gain. As for future work, we will consider applying this approach to a broader domain of CRS datasets. Currently, we only experiment on movie recommendations. However, the encoder of MESE is flexible and can potentially work well with cross-modality tasks. Multi-modal CRSs can be explored in the e-commerce domain with MESE.

References

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.

Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, page 191–198.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceddings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 1811–1818.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

Shirley Anugrah Hayati, Dongyeop Kang, Qingxiaoyang Zhu, Weiyan Shi, and Zhou Yu. 2020. INSPIRED: Toward sociable recommendation dialog systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8142–8152, Online. Association for Computational Linguistics.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, pages 173–182.

Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.

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 (NeurIPS)*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

Yu Li, Shirley Anugrah Hayati, Weiyan Shi, and Zhou Yu. 2021. DEUX: an attribute-guided framework for sociable recommendation dialog systems. *CoRR*, abs/2105.00825.

Zujie Liang, Huang Hu, Can Xu, Jian Miao, Yingying He, Yining Chen, Xiubo Geng, Fan Liang, and Daxin Jiang. 2021. Learning neural templates for recommender dialogue system. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7821–7833.

Ting Liu, Andrew W. Moore, Alexander Gray, and Ke Yang. 2004. An investigation of practical approximate nearest neighbor algorithms. In *Advances in Neural Information Processing Systems (NeurIPS)*, page 825–832.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 13–23.

Wenchang Ma, Ryuichi Takanobu, and Minlie Huang. 2021. Cr-walker: Tree-structured graph reasoning and dialog acts for conversational recommendation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1839–1851.

Marius Muja and David G Lowe. 2014. Scalable nearest neighbor algorithms for high dimensional data. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(11):2227–2240.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer.

Yue Shi, Martha Larson, and Alan Hanjalic. 2014. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Computing Surveys (CSUR)*, 47(1):1–45.

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 Thirty-first AAAI Conference on Artificial Intelligence*, page 4444–4451.

Xiaoyuan Su and Taghi M Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009.

Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. 2019a. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. *arXiv preprint arXiv:1908.01000*.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019b. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1441–1450.

Hao Tan and Mohit Bansal. 2019. LXMERT: Learning cross-modality encoder representations from transformers. 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 5100–5111, Hong Kong, China. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5998–6008.

Lingzhi Wang, Huang Hu, Lei Sha, Can Xu, Kam-Fai Wong, and Daxin Jiang. 2021. Finetuning large-scale pre-trained language models for conversational recommendation with knowledge graph. *arXiv preprint arXiv:2110.07477*.

Tianxing Wu, Arijit Khan, Huan Gao, and Cheng Li. 2019. Efficiently embedding dynamic knowledge graphs. *arXiv preprint arXiv:1910.06708*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online. Association for Computational Linguistics.

Guorui Zhou, Xiaoqiang Zhu, Chenru 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*, pages 1059–1068.

Jinfeng Zhou, Bo Wang, Ruifang He, and Yuexian Hou. 2021. CRFR: Improving conversational recommender systems via flexible fragments reasoning on knowledge graphs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4324–4334, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

A Candidate Selection With Multiple Sources

To maximize the performance of candidate selection, we blend candidates selected from the nearest neighbor search source with a rule based source. The rule based source is constructed by traversing the training data and maintaining a list of co-occurring items for each individual item. For example, if item A and item B both occurred in a conversation in the training set, we add them to each other's list. During testing, if item A is mentioned in the dialog context, the maintained co-occurring item list is combined with the candidates retrieved by nearest neighbor search as the final selection.

B Response Generation and Recommendation Trade-Off

The trade-off between response generation and recommendation during training. We separate language training objective from the jointly trained loss and compare the trained models on language generation metrics on ReDial in table 8. **MESE-LG** denotes the model trained with only the language generation objective $\mathcal{L}_{\text{res}}$.

Model	Dist2	Dist4	Bleu2	Bleu4
MESE-LG	**1.13**	**2.25**	**0.295**	**0.171**
MESE	0.822	1.313	0.246	0.143

Table 8: Results of **MESE** and **MESE-LG** on REDIAL.

As we can see, training only on language generation objective yields better results. In the settings of MESE, both tasks update the weights of the same pre-trained language model simultaneously with different goals: In recommendation, the language model needs to construct meaningful scores for generated candidates; In response generation, the language model aims to generate next token given previous contexts. This experiment showed that this joint training approach induces a trade-off between training on response generation and recommendation: back-propagation on the recommendation objective can lead to catastrophic forgetting of information learned from pre-training, which leads to worse language metrics.

C Response Generation Example

We include example dialog of human seeker and MESE model in table 9.

User: Hello, how are you?
System: Good, how are you?
What type of movies do you like?
User: I'm great. thanks!
I'm looking for crime movies to watch.
System: Have you seen **Goodfellas?**
User: Yes, I love that movie.
Could you recommend another one?
System: If you haven't seen **The Rise of The Krays**
you should give it a try.
User: I never saw that I should check it out.
Thank you and have a great afternoon!
System: You too! Have a great day!

Table 9: Conversation Example