AUGUST: an Automatic Generation Understudy for Synthesizing Conversational Recommendation Datasets

Yu Lu[2,1], **Junwei Bao**[3]*, **Zichen Ma**[2,1], **Xiaoguang Han**[1,2], **Youzheng Wu**[3], **Shuguang Cui**[1,2], **Xiaodong He**[3]
[1] SSE, CUHKSZ [2] FNii, CUHKSZ [3]JD AI Research
[2,1]{yulu1,zichenma1}@link.cuhk.edu.cn
[3]{baojunwei,wuyouzheng1,xiaodong.he}@jd.com
[1,2]{hanxiaoguang,shuguangcui}@cuhk.edu.cn

Abstract

High-quality data is essential for conversational recommendation systems and serves as the cornerstone of the network architecture development and training strategy design. Existing works contribute heavy human efforts to manually labeling or designing and extending recommender dialogue templates. However, they suffer from: (i) the limited number of human annotators results in that datasets can hardly capture rich and large-scale cases in the real world, (ii) the limited experience and knowledge of annotators accounts for the uninformative corpus and inappropriate recommendations. In this paper, we propose a novel automatic dataset synthesis approach that can generate both large-scale and high-quality recommendation dialogues through a data2text generation process, where unstructured recommendation conversations are generated from structured graphs based on user-item information from the real world. In doing so, we comprehensively exploit: (i) rich personalized user profiles from traditional recommendation datasets, (ii) rich external knowledge from knowledge graphs, and (iii) the conversation ability contained in human-to-human conversational recommendation datasets. Extensive experiments validate the benefit brought by the automatically synthesised data under the low-resource scenarios, and demonstrates the promising potential to facilitate developing a more effective conversational recommendation system[1].

1 Introduction

Conversational recommendation (CR) systems aim to recommend potential items of interest for users (or seekers) through dialogue-based interactions. Although tremendous works have been contributed to the CR domain, the lack of both large-scale and high-quality training data remains a common problem due to the great cost and difficulty in dataset construction. A classic recommendation dialogue collection (Li et al., 2018) relies on a human recommender to chat with a randomly paired seeker and supply some recommendations within several conversation turns usually based on the chatting content. The dataset constructed under this paradigm is not only limited in scale but also can hardly ensure the recommendation quality. Specifically, it suffers from: (i) the limited number of human annotators that results in datasets can hardly capture rich and large-scale cases in the real world, (ii) the limited experience and knowledge of annotators account for the uninformative corpus and inappropriate recommendations. In addition, the preference given by annotators to the recommended item may be "unreal" when (s)he is unfamiliar with it but cannot timely validate the annotation. The performance of a CRS trained with such datasets may be barely satisfactory when applied in real-world scenarios.

Although there exist numerous recommendation data that contain more "real-world" user preferences, *e.g.*, MovieLens (Harper and Konstan, 2015), there are little or even no corresponding dialogues, which leads to a low-resource scenario

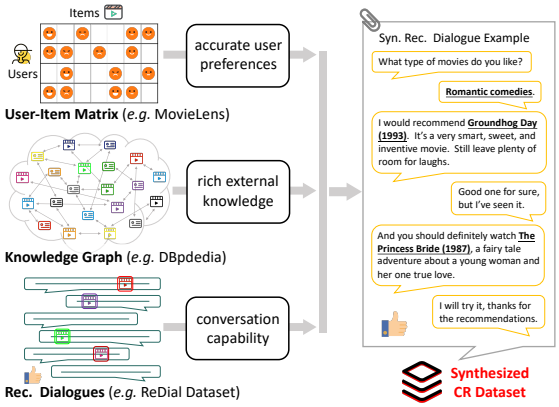

Figure 1: The proposed approach takes three kinds of sources, namely user-item matrices, knowledge graphs, and existing conversational recommendation datasets, to automatically generate recommendational dialogues.

*Corresponding author: baojunwei001@gmail.com
[1]Our code will be released in https://github.com/JD-AI-Research-NLP/AUGUST

Findings of the Association for Computational Linguistics: ACL 2023, pages 10538–10549
July 9-14, 2023

for CRS training. Therefore, we propose a novel CR data synthesis approach, AUGUST, which is an AUtomatic Generation UnderSTudy for conversational recommendation datasets. The core of our approach is synthesizing the strengths from three kinds of data resources: (i) user-item ratings from websites that can provide items really favored by each user; (ii) external knowledge that can provide rich item-related information leading to a "professional" recommender; (iii) abundant dialogue corpus that can help develop the learning model's conversation ability. Note that all three can be easily accessed thus facilitating the potential of generating large-scale diverse recommendational dialogues. In doing so, our approach contains two steps: (1) to form one data sample, seamlessly selecting some items rated by one user, from which a graph is constructed that contains the items, related entities, and their relations based on a well-developed knowledge graph (KG); (2) adopting a Data2Text generator (Li et al., 2021) to convert the item graph into a fluent and natural dialogue around the items. Such a graph-based dialogue generation manner is endowed with great extensibility and explainability where external knowledge can be integrated via expanding the intermediate graph with related entities from KG. To train the Data2Text module, we make use of recommendational dialogues from existing CR datasets to learn a dialogue generator. Specifically, we elicit graphs from dialogues as ones from user-item ratings, and train the Data2Text generator to take the graph as input to recover the original dialogue.

We conduct extensive experiments on the synthesized data quality and the performance of Data2Text generation, and give a detailed analysis of problems in the synthesis process. We also empirically validate the benefit of synthesized data in helping learn a stronger CRS, especially on recommendation accuracy in the low-resource scenario. Along with the rapid development of Data2Text generation methods, the proposed AUGUST is of great potential and provides a new solution to construct large-scale CR datasets, which is our main contribution. In addition, it is expected to attract more attention to the direction of automatic dataset generation, and facilitate the data-driven learning models designed for not only CR but also other various tasks in the future.

2 Related Work

2.1 Conversational Recommendation Dataset

Recently, Conversational Recommendation Systems (CRS) (Li et al., 2018; Chen et al., 2019; Jannach et al., 2021; Lu et al., 2021) have become an emerging research topic, which aims to provide high-quality recommendations to users through natural language. To facilitate the study of this task, some works collect human-human and human-machine conversation data by asking human annotators to conversate under certain rules. Hayati et al. manually annotate each utterance with the sociable strategies to validate the effectiveness of sociable recommendation strategies in CRS. Moon et al. present a parallel dialog↔KG corpus where each mention of an entity is manually linked with its corresponding KG paths. Liu et al. create a multi-type dialogue dataset and want the bots can proactively and naturally lead a conversation from a non-recommendation dialogue to a recommendation dialog. Similarly, Zhou et al. proposes a topic-guided CR dataset to help the research of topic transitions. However, Gao et al. point that existing datasets are not qualified to develop CRS that satisfies industrial application requirements for two reasons: 1) the scale of these datasets is not enough to cover the real-world entities and concepts; 2) the datasets constructed under certain rigorous constraints can hardly generalize to the complex and diverse real-world conversation. Therefore, more efforts are encouraged to develop large-scale, generalizable, and natural datasets for CRS.

2.2 Data2Text Generation

Data2Text Natural Language Generation (NLG) is the computational process of generating meaningful and coherent natural language text to describe non-linguistic input data. The input can be in various forms such as databases of records, spreadsheets, knowledge bases, and simulations of physical systems. Traditional methods for Data2Text generation (Reiter and Dale, 2000) implement a pipeline of modules including content planning, sentence planning, and surface realization. With the rapid development of Seq2Seq models especially pre-trained models, recent neural generation systems (Li et al., 2021) trained in an end-to-end fashion get state-of-the-art results on Data2Text benchmarks such as WebNLG (Gardent et al., 2017), ToTTo (Parikh et al., 2020), and AGENDA (Koncel-Kedziorski et al., 2019). One

of the most popular subtasks, Graph2Text, aims to create fluent natural language text to describe an input graph. Early works mainly center around statistical methods, applying grammar rule to generate text (Konstas and Lapata, 2013). Recently, neural-network-based approaches have been proposed to generate text from linearized KG triples (Ferreira et al., 2019), some of which investigate how to encode the graph structural information using Graph Neural Networks (GNNs) (Scarselli et al., 2008) and Transformer (Koncel-Kedziorski et al., 2019) explicitly. Unsupervised methods (Guo et al., 2020) and few-shot problems (Li et al., 2021) are also explored. In our approach, we adopt a Graph2Text generator for CR data synthesis.

3 Methodology

3.1 Preliminaries

Our CR dataset synthesis approach produces recommendational dialogues from three kinds of resources: user-item matrices from traditional recommendation datasets, external knowledge graphs, and existing CR datasets. We first introduce related notations. **A user-item matrix** (UIM) $\mathbf{M}$ (supplied by datasets like MoviLens (Harper and Konstan, 2015)) consists of N rows and M columns, of which the i-th row represents the ratings of the i-th user U_i towards all M items, and each element $s_{ij} \in [1, 2, 3, 4, 5]$ represents the i-th user's rating score towards the j-th item o_j, where a higher score represents the user's more favor to one item. Note that the matrix $\mathbf{M}$ may be sparse depending on the number of ratings given by each user. **A knowledge graph** $\mathbf{G} =< \mathcal{E}, \mathcal{R} >$, *e.g.* DBpedia (Auer et al., 2007), where $\mathcal{E}$ and $\mathcal{R}$ are the entity and relation set, respectively. The graph consists of large amounts of entity-relation-entity triples (e_i, r_{ij}, e_j), of which e_i or e_j can be an item or non-item entity from $\mathcal{E}$ and $r_{ij} \in \mathcal{R}$ represents the relation category between an associated entity pair. We denote the item entity set as $\mathcal{O} \subset \mathcal{E}$, which contains all recommendation candidates. In **a CR dataset**, *e.g.* the ReDial dataset (Liu et al., 2020), a conversation is generated for recommendations on a certain domain (movie, traveling, or restaurant, etc.) in a seek-recommender pair. Denote the i-th conversation as C_i, a seeker/user U_i is asking for item recommendations from a recommender R_i. In the following chatting turns, U_i may express his/her preferences explicitly or implicitly, then R_i is expected to capture the user's preferences according to the historical dialogue context, denoted as $\mathcal{C}_t = \{c_j\}_1^t$, where t is the historical turn number and c_j is the j-th conversation utterance.

3.2 Dataset Synthesis

The proposed dataset synthesis approach starts from real-world user preferences information easily accessed from the UIM $\mathbf{M}$. Then a UIM→Graph→Dialogue generation pipeline is adopted to synthesize recommendational dialogues, with the overview shown in Fig. 2.

UIM → Graph The first step is to convert UIM that contains user preferences into graphs. From any row i of $\mathbf{M}$, a set of items with respective ratings $\{(o_j, s_{ij})\}$ can be taken to generate a dialogue sample. All o_j are used as nodes to construct the graph $\mathbf{G}'_i$. To integrate the user preferences into $\mathbf{G}'_i$, an extra node of user u_i with its relation to each item node is added to constitute triples like (u_i, s_{ij}, o_j) for item o_j. Furthermore, we extend $\mathbf{G}'_i$ by incorporating rich external knowledge from $\mathbf{G}$ for the informativeness of the final dialogue output. Specifically, for each two items o_j and o_k, we search for a two-hop path in $\mathbf{G}$ to find their relations, *i.e.*, two movies are directly linked (neighbouring) as (o_j, r_{jk}, o_k) (*e.g.* belong to one movie series) or linked by one entity e_l as $(o_j, r_{jl}, e_l, r_{lk}, o_k)$ (*e.g.* sharing the same director, actors, or genre). Then, these triples in the searched paths are added into $\mathbf{G}'_i$. The obtained graph $\mathbf{G}'_i$ can better represent the selected items from UIM data by incorporating both accurate user-preference information and knowledge-equipped inter-entity relations.

Graph → Dialogue Given a graph $\mathbf{G}'_i$ that represents the items expected to appear in the dialogue, a Data2Text generator aims to synthesize a conversational dialogue C_i based on the graph. We cast it as a Data2Text problem. We adopt a Data2Text generator to take the graph as input, and output raw text that contains the vertex and edge information in the graph. Note that two tokens [U] (user) and [R] (responder) are specially defined to be generated in the text, such that the sentences after [U] ([R]) and before the next token [R] ([U]) can be viewed as a single turn. In this way, the text can be decomposed and re-organized into a multi-turn dialogue. Considering there is no supervision (graph-dialogue data pair) for the learning of the generator in this Data2Text process, we utilize the conversation corpus in existing CR datasets to learn

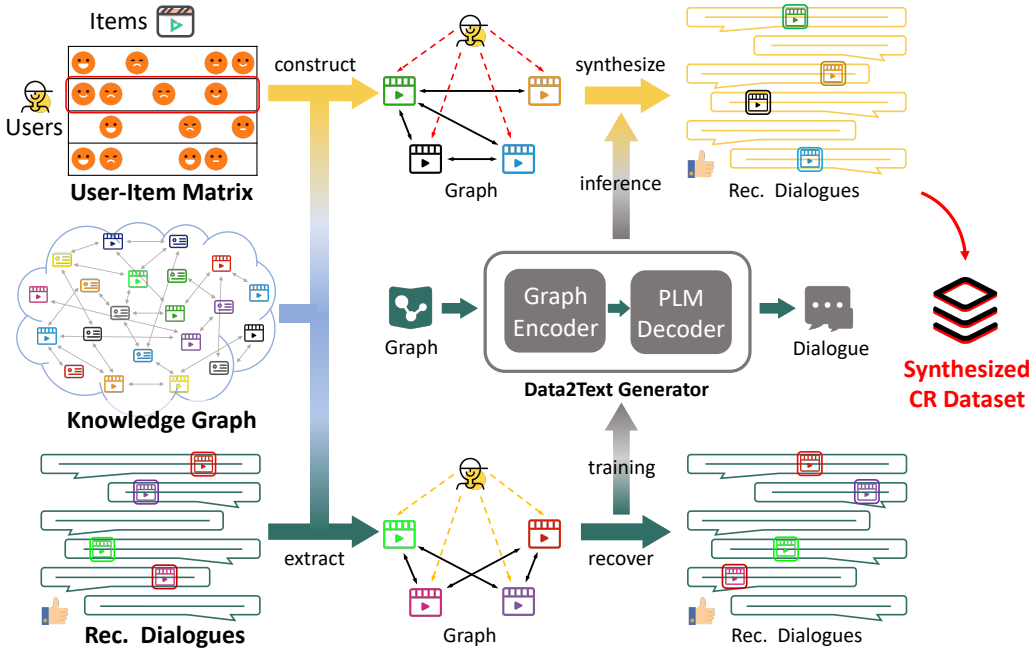

Figure 2: The overview of the proposed AUGUST framework for automatic recommendational dialogue synthesis.

a strong generator for dialogue synthesis, which is introduced in the following subsection.

3.3 Data2Text Generation

In order to generate both natural and logical dialogues from item-related graphs, we adopt a Data2Text generator to learn the conversation knowledge in existing CR datasets for Graph $\rightarrow$ Dialogue generation. As illustrated in Fig. 3, an encoder-decoder architecture is implemented with an R-GCN encoder (Schlichtkrull et al., 2018) for graph feature extraction, and a pre-trained language model (PLM) (Lewis et al., 2020) decoder for dialogue generation.

Graph Construction and Encoding Given any dialogue sample C_i in existing CR datasets, we construct a graph $\mathbf{G}'_i$ to produce a graph-dialogue training pair for learning a strong Data2Text generator. To construct $\mathbf{G}'_i$ from C_i, we first search for all entities $\{e_j\}$ with the speaker's (U_i or R_i) sentiment $\{s_{ij}\}$ to them (provided by CR datasets usually or generated from an estimator), and link each e_j with corresponding nodes in $\mathbf{G}$. Then a graph $\mathbf{G}'_i$ can be constructed in a similar way as in the UIM $\rightarrow$ Graph process described in Sec. 3.2. Given a constructed $\mathbf{G}'_i$, an R-GCN (Schlichtkrull et al., 2018) is applied as the encoder to generate entity embeddings for $\mathbf{G}'_i$. Let $\phi_j \in \mathbb{R}^d$ denote the entity embedding for a general entity e_j in KG, where d is the embedding size. Then the R-GCN helps leverage the multi-relational information to have a structure-aware graph representation. Specifically, the embedding of e_j at the $l+1$-th of total L layers can be computed as:

$$\phi_j^{l+1} = \sigma\Big(\sum_{r\in\mathcal{R}}\sum_{k\in\mathcal{N}_j^r} \mathbf{W}_r^l \phi_k^l + \mathbf{W}_0^l \phi_j^l\Big),$$

where $\sigma(\cdot)$ is the activation function, $\mathbf{W}_r^l$ and $\mathbf{W}_0^l$ are trainable parameters, and $\mathcal{N}_j^r$ is the set of neighbouring entities of e_j under relation r. Note that, all ϕ_j^0 before the first layer are initialized by pre-trained KG embeddings in (Yang et al., 2014). The entity embeddings $\{\phi_j^L\}$ output by the last R-GCN layer are re-denoted as $\{\phi_j\}$ for simplification.

Graph Feature Learning To learn higher-quality graph features for more smooth decoding, we leverage another encoding branch of a pre-trained language model (PLM) to learn context-aware node features and align ones encoded from graphs with them. Specifically, by taking the whole dialogue as PLM input, entities are represented with contextual information in natural utterances, so that rich knowledge in PLM can be adapted. Denote the context-aware entity embedding output by the PLM branch as $\hat{\phi}_j \in \mathbb{R}^d$, which has the same dimension as the R-GCN embedding. The alignment between two types of entity feature vectors is implemented by minimizing an l_2 loss, denoted as

L_{align}:

$$L_{align} = \sum_{e_j \in \mathbf{G}'_i} ||\phi_j - \hat{\phi}_j||^2.$$

Before feeding graph node features into the decoder, we linearize them into an entity sequence $\{\phi_j\}$ through a relation-biased breadth-first search (RBFS) strategy following (Li et al., 2021), where a breadth-first search is adapted and an RBFS weight α_j is computed for each node e_j as its score to decide the order in each search level:

$$\alpha_j = \sigma(\phi_i^\top \mathbf{W}_r^L \phi_j), < e_i, r, e_j > \in \mathbf{G}',$$

where e_i is the parent node of e_j in the search process. In the same search level, the node with a higher RBFS score has a higher order in the sequence. For more related implementation details, please refer to (Li et al., 2021).

Dialogue Decoding In the decoding stage, a PLM decoder is performed to decode the linearized graph features $\{\phi_j\}$ into textual dialogues. To formalize the dialogue generation into a typical natural language generation problem, we sequentially connect all utterances into a single paragraph but with special tokens as the separation for regrouping into dialogue turns. Denote the k-th of total K tokens as w_k, the generation objective is to minimize the negative log-likelihood as:

$$L_{gen} = -\sum_{k=1}^{K} \log P(w_k | w_1, w_2, \cdots, w_{k-1}),$$

where $P(\cdot)$ denotes the probability function. To encourage covering entities from the input graph, a copy mechanism implemented with a pointer network is conducted, leading to a copy loss term L_{copy}.

The overall objective function to learn the domain adaptive encoder-decoder can be written as:

$$L_{over} = L_{gen} + \lambda_1 L_{align} + \lambda_2 L_{copy},$$

where λ_1 and λ_2 are weight factors to balance different loss terms, respectively.

4 Experiments

4.1 Experiment Setting

4.1.1 Resources

(1) The ReDial dataset (Li et al., 2018) is collected by crowd-sourcing users on Amazon Mechanical Turk (AMT). Two paired workers serve as the recommender and user to produce a conversation and cover at least 4 different movies. Every movie mentioned in the dialog is annotated explicitly. ReDial contains 10,021 conversations related to 64,362 movies and is split into training, validation, and test sets with a ratio of 8:1:1. **(2) The MovieLens** dataset (Harper and Konstan, 2015), released by GroupLens Research, describes people's expressed preferences for movies. These preferences take the form of <user, item, rating, time-stamp> tuples, where the rating (1∼5) represents the user's preference for a movie at a particular time. These preferences are collected by the MovieLens website, a recommender system that asks its users to give movie ratings for personalized movie recommendations. **(3) The DBpedia knowledge base** (Auer et al., 2007) contains structured knowledge extracted from Wikipedia. It collects rich movie-related information and inter-movie relations and releases an open knowledge graph available to the public.

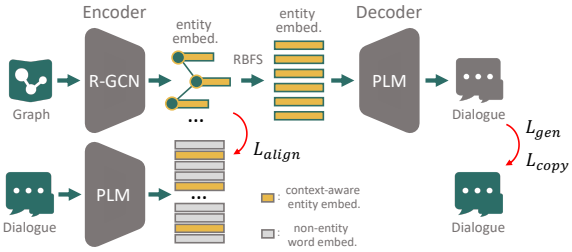

Figure 3: The illustration of the used encoder-decoder architecture for Data2Text generation.

4.1.2 Datasets

To validate the Data2Text generation quality of AUGUST, we construct graph-dialogue pairs from the ReDial (Li et al., 2018) and WebNLG (Gardent et al., 2017) dataset for training and evaluation. Considering the limitations of existing datasets as stated in Sec. 1, we create a small dataset with more "real-world" and reliable recommendations for CR evaluation. We sample 200 pieces of user-item data from MovieLens and hire some annotators to create conversations according to the user preferences for the movies, named "ML-G2D" in Tab. 1. We also provide annotators with external knowledge (*e.g.*, movie websites) and ReDial dialogue samples as references to guarantee conversation quality. Among the annotated 200 dialogues, 100 are randomly sampled and used for training in the low-resource scenario, and the other 100 are set as the test set. Note that when testing on WebNLG in

Test Data	Reconstruction						Writing Quality			
	Recall	**B-2**	**B-4**	**R-L**	**CIDEr**	**Chrf**	**Dist-1**	**Dist-2**	**Dist-3**	**PPL**
WebNLG	-	35.01	19.82	48.02	1.65	43.65	0.90	0.92	0.87	9.16
ReDial	0.82	28.44	11.43	28.82	1.52	42.16	0.43	0.75	0.84	3.39
ML-G2D	0.78	21.51	7.53	24.82	1.04	32.60	0.44	0.84	0.98	3.57

Table 1: Performance of Data2Text generation on three datasets. B-n denotes BLEU-n and R-L denotes ROUGE-L.

Data	Distinct.		Language Nat.		
	Dist-2	**Dist-3**	**Logic**	**Fluency**	**Inform.**
AUGUST	2.7	4.2	4.4	4.0	3.9
ReDial	2.8	4.4	4.6	4.6	4.0

Table 2: Comparison on Distinctness and Language Naturalness (via human evaluation) of AUGUST synthesized data and ReDial data. "Inform." means informativeness.

Tab. 1, we use WebNLG as the dialogue resource to train the Data2Text generator in AUGUST, and when testing on ReDial and ML-G2D, we both use ReDial as the dialogue resource. To validate the benefit of synthesized data by our AUGUST, we implement experiments to use our synthesized data as training data for the CR task. Note that to compare the benefit brought by the synthesized data and ReDial data, we randomly sample around 8,000 pieces from the synthesized data for the later training of KGSF, which keeps the same scale as ReDial training data. The synthesized data is denoted as "AUGUST" in Tab. 3 and 4.

4.1.3 Evaluation Metrics

To investigate the performance of various methods on the Data2Text generation task, we first conduct evaluations on the quality of **conversation reconstruction**. We adopt four automatic evaluation metrics widely used in Data2Text generation tasks (Li et al., 2021): BLEU (Papineni et al., 2002) and ROUGE-L (Lin, 2004), which computes the overlap ratio of n-grams between the reconstructed dialogue and the original one; CIDEr (Vedantam et al., 2015) that computes the TF-IDF weights for each n-gram in synthetic/real dialogues; and Chrf++ (Popović, 2017) that computes the average F-score on both character-level and word-level n-grams. In addition, we also compute the recall ratio (Recall) of entities to measure how many entities are recovered in the dialogue relative to the graph input. For the **conversation writing** quality, we compute Dist-n (Li et al., 2015) to show the distinctness of the generated utterances and the perplexity (PPL) proposed in (Jelinek et al., 1977) to measure the language fluency. Besides, we also conduct human evaluation to show the generation quality following the previous works in (Li et al., 2021; Agarwal et al., 2021), which contains three workers' ratings to 200 randomly sampled dialogues with respect to language naturalness including aspects of fluency, dialogue logic, and informativeness (5 is the full score). As for the **evaluation of CRS** trained on the synthesized data by AUGUST, we follow (Li et al., 2018; Chen et al., 2019; Zhou et al., 2020a) to use Recall@k (R@k, $k = 1, 10, 50$) as the recommendation evaluation metric, which indicates whether the predicted top-k items contain the ground truth recommendation provided by human recommenders. The generation quality of CRS is evaluated on Dist-n and PPL as in the Data2Text generation task.

4.1.4 Implementation Details

In the step of Data2Text generation, the graph encoder in AUGUST is implemented as a two-layer R-GCN with an embedding size of 1,024. The PLM encoder for context-aware entity embedding adopts the encoder of a pre-trained BART-large (Lewis et al., 2020), which is a transformer-based model with a bidirectional encoder and an autoregressive decoder. The initial weights are provided by Hugging Face[2] and are frozen in training. As for the text decoder, we employ the decoder of a BART-large initialized with pre-trained weights for dialogue generation. The parameters in the R-GCN encoder and BART decoder are optimized using an AdamW (Loshchilov and Hutter, 2017) optimizer with a learning rate of 10^{-5}. The weight factors, λ_1 and λ_2, are set to 0.8 and 0.8, respectively. The whole network is trained on 4×23GB NVIDIA Tesla P40 with a minibatch size of 16. To validate

[2]https://huggingface.co/facebook

ReDial	AUGUST	ML	Recommendation			Conversation			
			R@1	R@10	R@50	Dist-2	Dist-3	Dist-4	PPL
		0%	0.00	0.01	0.02	-	-	-	-
		50%	0.08	1.76	2.21	-	-	-	-
		100%	0.17	1.59	2.94	-	-	-	-
✓		0%	0.00	1.77	3.53	0.292	0.336	0.470	14.2
✓		50%	0.00	2.65	6.19	0.303	0.411	0.482	15.7
✓		100%	0.01	2.77	6.02	0.321	0.374	0.510	16.9
	✓	0%	1.32	4.42	15.93	0.239	0.315	0.318	**11.4**
	✓	50%	**1.76**	4.42	14.16	0.307	0. 316	0.425	14.6
	✓	100%	0.88	**7.79**	**19.46**	0.297	0.301	0.412	13.8
✓	✓	0%	0.17	1.77	8.65	0.292	0.375	0.451	14.0
✓	✓	50%	0.84	2.66	10.31	**0.360**	**0.451**	0.507	15.7
✓	✓	100%	0.91	3.54	9.84	0.318	0.445	**0.522**	16.0

Table 3: Performance on ML-G2D test set when incorporating different types of training data, including **ReDial** training data, **AUGUST** synthesized data, and **ML**-G2D training set.

ReDial	AUGUST	R@1	R@10	R@50
		0.0	0.0	0.0
	✓	2.5	15.7	33.2
✓		3.9	18.3	37.8
✓	✓	3.2	17.8	36.6
F	P	**5.3**	**25.1**	**47.1**

Table 4: Recommendation accuracy on the ReDial test set when trained on the ReDial and AUGUST data.

the benefit of synthesized data by AUGUST, we implement a popular CRS, KGSF (Zhou et al., 2020a), as the baseline, which incorporates two KGs, ConceptNet (Speer et al., 2017) and DBpedia (Auer et al., 2007), to enhance the data representations. Implementation details can be referred to in the released codes by Zhou et al.[3].

4.2 Experiment Results

4.2.1 Data2Text Evaluation

We give both automatic and human evaluations of the generation quality by AUGUST. For automatic evaluation, we implement AUGUST with BART-large as the PLM, on all three datasets to construct a benchmark for future related works. As shown in Tab. 1, with the same training data, AUGUST performs poorer on ML-G2D than on Re-G2D, which may result from the distribution bias of ReDial data with real-world user preferences as stated in Sec. 1. Besides, the PPL values are low in all settings, so the generation has high confidence, which may result from the consistency of the generation objective between BART pre-training and Data2Text training. Performances on WebNLG are higher than on the other two over all metrics except PPL, because the target text in WebNLG is usually shorter and with richer common entities, and the input has fewer triples, which reduces the generation difficulty. Besides, we also directly compare the quality of the synthesized data by AUGUST and the ReDial data, on "Distinctness" and "Language Naturalness" in Tab. 2. We compute the Dist-2 and Dist-3 scores, and conduct human evaluation on the dialogue logic, fluency, and informativeness, which shows that the synthesized data has a high quality that is close to the ReDial data on both utterance distinctness and language naturalness.

4.2.2 CR Evaluation

We evaluate the CR performance of KGSF on the ML-G2D test set, with using different types of data in training. The training data is a combination of external data: ReDial training set and AUGUST synthesized data, and internal data: ML-G2D training set. We set the ratio of the used ML-G2D data to 0%, 50%, and 100% to investigate the performance in low-resource scenarios with different extents. From the results in Tab. 3, it can be seen: (i) KGSF without any external training data (ReDial or AUGUST) performs poor on recommendation; (ii)

[3]https://github.com/Lancelot39/KGSF

Using only ReDial as external data can bring benefits to the conversation generation, but leads to only a tiny improvement in recommendation; (iii) Using only AUGUST as external data can bring a significant improvement in recommendation compared with ReDial, especially in a more low-resource scenario; (iv) Using both ReDial and AUGUST as external data cannot bring extra gains on recommendation accuracy but can improve the distinctness in the generated conversations. These results show that: (1) There exists a distribution bias between the recommendations in ReDial data and the user preference in the real world, which results in the unsatisfying recommendation performance of a ReDial-trained CRS; (2) The synthesized data by AUGUST is useful to help a KGSF capture real-world user preferences for conversational recommendations, especially in low-resource scenarios; (3) ReDial and AUGUST data are complementary to provide a more rich corpus for improving the conversation capability of CRS, and adding AUGUST data also leads to a higher recommendation accuracy than using ReDial data only.

We also evaluate the recommendation performance of KGSF on the ReDial test set when trained with ReDial or/and AUGUST data. As shown in Tab. 4, it can be seen: (i) The recommendation accuracy of KGSF is low without any training data; (ii) Adding synthesized AUGUST data can bring performance gain to get close to but lower than adding real ReDial training data; (iii) Simply adopting joint training with ReDial and AUGUST data can only obtain similar performance as using ReDial data only; (iv) Using AUGUST data as pre-training and finetuning on ReDial data can bring an extra performance gain. The results of (ii) further prove the benefit of synthesized data by AUGUST and the distribution bias between ReDial recommendations and real-world user preferences. In addition, although simply jointly using both data for training can hardly bring performance gain as in (iii) considering the distribution bias, the synthesized AUGUST data can still help improve the recommendation ability of KGSF when using AUGUST data for pre-training and finetuning on ReDial data. In this way, the AUGUST data provide a better initialization for the optimization of KGSF, and finetuning on ReDial data can guarantee the distribution consistency. It also shows the great potential of AUGUST to serve as a data synthesis approach for a better initialization of parameters in CRS.

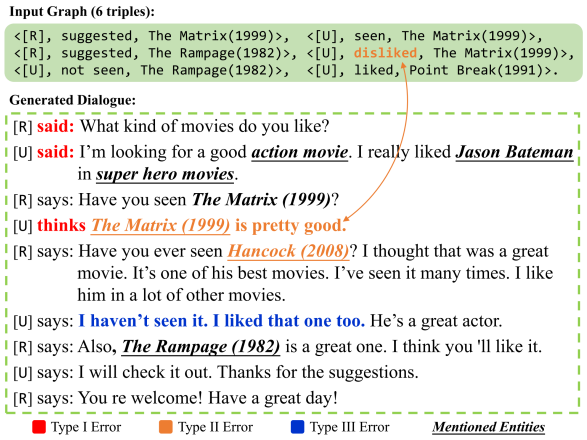

Figure 4: Visualization of a generation case by AUGUST for error analysis.

4.2.3 Error Analysis

We summarize three types of errors that appeared in our generation according to the hierarchy of the dialogue requirement, with one example shown in Fig. 4. **Error Type I: Format Errors**, including grammar and spelling mistakes, or the unexpected writing format, *e.g.*, each utterance is expected to start with the identity of "[U] says:", while it may generate "[U] thinks". **Error Type II: Hallucination**, which is a common problem in language generation tasks. It means the network (i) generates contents that conflict with the input data, *e.g.* producing wrong relations, entities, or sentiments, or (ii) generates extra items beyond the input, which means the output is not a precise description to the input, *e.g.* "Hancock (2008)" in Fig. 4. **Error Type III: Incoherent Logic**, which refers to the problem of incoherent or contradictory logic in the generated dialogue, *e.g.* the user says (s)he has not seen a movie but liked it.

5 Conclusion

This paper proposes an automatic generation understudy for conversational recommendation datasets. By casting the dialogue synthesis process as a Data2Text generation task, a baseline framework is constructed to exploit (i) rich accurate user preferences from user-item matrices, (ii) rich external knowledge from external knowledge graphs, and (iii) the conversation ability from the corpus of existing CR datasets. Experiment results show that our generation is comparable to human-labeled conversations and superior in scalability, extensibility, and explainability. More importantly, we empirically show the benefit of our synthesized data in

improving a CRS, especially in recommendation accuracy. The proposed approach exhibits great potential for automatic dataset synthesis and is expected to inspire researchers in other fields.

Limitations

The limitations of this work mainly lie in two aspects: (i) The synthesis quality is determined by the performance of existing Data2Text approaches, while Data2Text generation is still a difficult task that waiting for deeper exploration. The common errors in generation are included in Sec. 4.2.3. (ii) We adopt a PLM as the decoder in Data2Text generation in order to generate fluent utterances. However, as stated in (Ribeiro et al., 2021), PLMs tend to pay more attention to sentence fluency than to the graph structures of inputs, which may cause the loss of some critical information.

Acknowledgement

The work was supported in part by NSFC with Grant No. 62293482, the Basic Research Project No. HZQB-KCZYZ-2021067 of Hetao Shenzhen-HK S&T Cooperation Zone, the National Key R&D Program of China with grant No. 2018YFB1800800, the Shenzhen Outstanding Talents Training Fund 202002, the Guangdong Research Projects No. 2017ZT07X152 and No. 2019CX01X104, the Guangdong Provincial Key Laboratory of Future Networks of Intelligence (Grant No. 2022B1212010001), the Shenzhen Key Laboratory of Big Data and Artificial Intelligence (Grant No. ZDSYS201707251409055), and the National Key R&D Program of China under Grant No. 2020AAA0108600.

References

Oshin Agarwal, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. 2021. Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3554–3565.

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.

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.

Thiago Castro Ferreira, Chris van der Lee, Emiel Van Miltenburg, and Emiel Krahmer. 2019. Neural data-to-text generation: A comparison between pipeline and end-to-end architectures. 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 552–562.

Chongming Gao, Wenqiang Lei, Xiangnan He, Maarten de Rijke, and Tat-Seng Chua. 2021. Advances and challenges in conversational recommender systems: A survey. *AI Open*, 2:100–126.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The webnlg challenge: Generating text from rdf data. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133.

Qipeng Guo, Zhijing Jin, Xipeng Qiu, Weinan Zhang, David Wipf, and Zheng Zhang. 2020. Cyclegt: Unsupervised graph-to-text and text-to-graph generation via cycle training. In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 77–88.

F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19.

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.

Dietmar Jannach, Ahtsham Manzoor, Wanling Cai, and Li Chen. 2021. A survey on conversational recommender systems. *ACM Computing Surveys (CSUR)*, 54(5):1–36.

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.

Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. 2019. Text generation from knowledge graphs with graph transformers. 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 2284–2293.

Ioannis Konstas and Mirella Lapata. 2013. Inducing document plans for concept-to-text generation. In *Proceedings of the 2013 Conference on Empirical*

Methods in Natural Language Processing, pages 1503–1514.

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. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

Junyi Li, Tianyi Tang, Wayne Xin Zhao, Zhicheng Wei, Nicholas Jing Yuan, and Ji-Rong Wen. 2021. Few-shot knowledge graph-to-text generation with pretrained language models. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1558–1568.

Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards deep conversational recommendations. *Advances in neural information processing systems*, 31.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

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.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

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.

Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. 2019. Opendialkg: Explainable conversational reasoning with attention-based walks over knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 845–854.

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.

Ankur Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. Totto: A controlled table-to-text generation dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1173–1186.

Maja Popović. 2017. chrf++: words helping character n-grams. In *Proceedings of the second conference on machine translation*, pages 612–618.

Ehud Reiter and Robert Dale. 2000. *Building Natural Language Generation Systems*. Cambridge University Press.

Leonardo FR Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2021. Investigating pretrained language models for graph-to-text generation. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 211–227.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80.

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.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

Kun Zhou, Wayne Xin Zhao, Shuqing Bian, Yuanhang Zhou, Ji-Rong Wen, and Jingsong Yu. 2020a. 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.

Kun Zhou, Yuanhang Zhou, Wayne Xin Zhao, Xiaoke Wang, and Ji-Rong Wen. 2020b. Towards topic-guided conversational recommender system. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4128–4139.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Left blank.

☑ A2. Did you discuss any potential risks of your work?
Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Left blank.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☒ Did you use or create scientific artifacts?

Left blank.

☐ B1. Did you cite the creators of artifacts you used?
Not applicable. Left blank.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Not applicable. Left blank.

C ☑ Did you run computational experiments?

Left blank.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Left blank.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Left blank.

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Left blank.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Left blank.

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

☒ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Left blank.

☒ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Left blank.

☑ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Left blank.

☒ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Left blank.