Knowledge-grounded Natural Language Recommendation Explanation

Anthony Colas[*1], **Jun Araki**[2], **Zhengyu Zhou**[2],
Bingqing Wang[2], **Zhe Feng**[2]
[1]University of Florida
[2]Bosch Research North America
acolas1@ufl.edu
{jun.araki, zhengyu.zhou2, bingqing.wang, zhe.feng2}@us.bosch.com

Abstract

Explanations accompanying a recommendation can assist users in understanding the decision made by recommendation systems, which in turn increases a user's confidence and trust in the system. Recently, research has focused on generating natural language explanations in a human-readable format. Thus far, the proposed approaches leverage item reviews written by users, which are often subjective, sparse in language, and unable to account for new items that have not been purchased or reviewed before. Instead, we aim to generate fact-grounded recommendation explanations that are objectively described with item features while implicitly considering a user's preferences, based on the user's purchase history. To achieve this, we propose a knowledge graph (KG) approach to natural language explainable recommendation. Our approach draws on user-item features through a novel collaborative filtering-based KG representation to produce fact-grounded, personalized explanations, while jointly learning user-item representations for recommendation scoring. Experimental results show that our approach consistently outperforms previous state-of-the-art models on natural language explainable recommendation metrics.[1]

1 Introduction

Current approaches to natural language (NL) explainable recommendation focus on generating user reviews (Chen et al., 2018; Wang et al., 2018a; Li et al., 2020, 2021; Yang et al., 2021). Instead of providing a justification for the item recommendation, the models learn to output language that is commonly found in personal reviews. This reliance on reviews poses three problems: 1) The explanations are not objective, because users typically review items based on their sentiment (Wu et al., 2018), 2) Reviews are often sparse, because they describe a user's own experience (Asghar, 2016), 3) Systems that rely on reviews cannot account for new items which have never been purchased before, nor can they provide justifications for item catalogs which may not have reviews available. Given this, it may be difficult for a user to reason as to why an item was recommended, hindering the user's experience (Tintarev and Masthoff, 2015). The user may then lose trust in such systems which do not provide objective and accurate explanations.

We propose **KnowRec**, a KG-grounded approach to natural language explainable recommendation which not only personalizes recommendations/explanations with user information, but also draws on facts about a particular item via its corresponding KG to generate objective, specific, and data-driven explanations for the recommended item. For example, given the movie "Paths of Glory", previous work aims to generate explanations such as "it's not the best military movie" and "good performances all around", which are subjective, not specific to a given movie, and relies on data from pre-existing reviews. Instead, by leveraging an item KG such as *<director, Stanley Kubrick>, <conflict, World War 1>, <country, France>*, a more objective and precise explanation can be produced such as: "A World War I French colonel defends three soldiers. Directed by Stanley Kubrick." The item features of 'World War I', 'colonel', and 'defends three soldiers' in the explanation objectively describe the movie, while they can implicitly reflect the user's preferences for war movies, based on his/her purchase history.

KnowRec is also more advantageous than prior work in terms of scalability to unpurchased items. Previously, KG-based recommendation systems have effectively addressed the cold-start problem by linking users and items through shared attributes (Wang et al., 2019, 2020, 2021). Similarly, there exists a kind of cold-start problem for new

[*] Work performed at Bosch Research.

[1] Our code and datasets are available at: https://github.com/boschresearch/KnowRec.

Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP, pages 1–15
December 7, 2023.

items in recommendation explanation that rely on reviews. KnowRec demonstrates KGs can help solve this problem through existing item-level features by adapting KG-to-text (Koncel-Kedziorski et al., 2019; Ke et al., 2021; Colas et al., 2022) elements into explainable recommendation, producing item-level explanations to justify a purchase. The KG-based approach is particularly important for recommendation scenarios in special domains where personal reviews are not available and the review-based approaches are impractical.

Our approach presents several algorithmic novelties. First, inspired by work on KG Recommendation (Wang et al., 2020) and KG-to-Text (Colas et al., 2022), we devise a novel user-item KG lexical representation, viewing the input through collaborative filtering lens, where users are graphically represented via their previous purchases and connected to a given item KG. Our representation differs from previous work on explainable NL generation which relies on ID and sparse keyword features. Previous work extracts keywords from reviews to represent the user and item, linearizing all such features to encode and produce an NL explanation (Li et al., 2020, 2022). Next, KnowRec adapts a graph attention encoder for the user-item representation via a new masking scheme. Finally, the encoded KG representation is simultaneously decoded into a textual explanation, while we innovatively dissociate the joint learned user-item representation to compute a user-item similarity for recommendation scoring.

To evaluate our approach, we first devise a method of constructing $(KG, Text)$ pairs from product descriptions as described in Section 5, where we extract entities and relations for the item KGs. We construct two such datasets from the publically available recommendation datasets to evaluate our proposed model for both the explanation and recommendation task and focus on natural language generation (NLG) metrics for the explanation task as in previous work. We adapt and compare previous baseline models for the recommendation explanation task as described in Section 6, where we substantially outperform previous models on explanation while achieving similar recommendation performance as models that rely on user and item ID-based features.

2 Related Work

2.1 Explainable Recommendation

Previous works on NL explainable recommendation focus on generating user-provided reviews, where the output is typically short, subjective, and repetitive (Chen et al., 2018; Hou et al., 2019; Wang et al., 2018b; Yang et al., 2021; Li et al., 2017, 2020, 2021; Hui et al., 2022). Extractive-based approaches have been proposed to score and select reviews as explanations (Chen et al., 2018; Li et al., 2019). Conversely, generative approaches (Yang et al., 2021; Li et al., 2017, 2020, 2021; Sun et al., 2020; Hui et al., 2022) leverage user/item features to generate new reviews as explanations. Currently, the task is still limited by review data, thus these models cannot adequately handle new items. Unlike previous work, we introduce KGs to the explainable recommendation task to provide objective, information-dense, specific explanations. Our approach can then handle new items which have not been reviewed yet.

Inspired by recent advancements in explainable recommendation models like (Li et al., 2021), we enhance BART (Lewis et al., 2020), renowned for graph-to-text tasks, to incorporate user-item knowledge graphs. This adaptation enables us to generate recommendation scores along with natural language explanations.

2.2 Knowledge Graph Recommendation

Leveraging KGs for recommendation systems has gained increasing attention (Wang et al., 2019, 2020, 2021; Xie et al., 2021; Du et al., 2022). In neighborhood-based methods (Hamilton et al., 2017; Welling and Kipf, 2016; Veličković et al., 2018), propagation is performed iteratively over the neighborhood information in a KG to update the user-item representation. While recent work has produced explanations via KGs, these works focus on structural explanations such as knowledge graph paths (Ma et al., 2019; Fu et al., 2020; Xian et al., 2019) and rules (Zhu et al., 2021; Chen et al., 2021; Shi et al., 2020), which are not as intuitive for users to understand. We focus on generating NL explanations, which has been shown to be a preferred type of explanation (Zhang et al., 2020). For a fair comparison, we compare to prior work that produces NL explanations. Unlike these works, we aim to generate NL explanations instead of using paths along the KG as explanations.

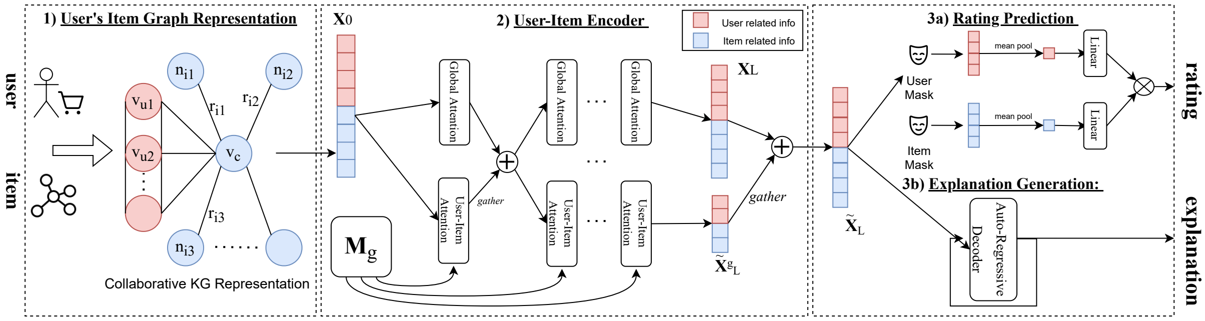


Figure 1: Illustration of KnowRec. 1) The User's Item KG Representation Module. 2) The Global and User-Item Graph Attention Encoder. 3) The Output Module for rating prediction and explanation.

2.3 Knowledge Graph-to-Text Generation

In KG-to-Text, pre-trained language models such as GPT-2 (Radford et al., 2019) and BART (Lewis et al., 2020) have seen success in generating fluent and accurate verbalizations of KGs (Chen et al., 2020; Ke et al., 2021; Ribeiro et al., 2021; Colas et al., 2023). We devise an encoder for user-item KGs and a decoder for both the generation and recommendation tasks. Specifically, we formulate a novel masking scheme for user-item KGs to structurally encode user and item features, while generating a recommendation score from their latent representations. Thus, our task is two-fold, fusing elements from the Graph-to-Text generation and KG recommendation domains.

3 Problem Formulation

Following prior work, we denote $\mathcal{U}$ as a set of users, $\mathcal{I}$ as a set of items, and the user-item interaction matrix as $\mathbf{Y} \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{I}|}$, where $y_{uv} = 1$ if user $u \in \mathcal{U}$ and item $v \in \mathcal{I}$ have interacted. Here, we represent user u as the user's purchase history $u = \{v_{ui}\}$, where v_{ui} denotes the i-th item purchased by user u in the past. Next, we define a KG as a multi-relational graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of entity vertices and $\mathcal{E} \subset \mathcal{V}\times\mathcal{R}\times\mathcal{V}$ is the set of edges connecting entities with a relation from $\mathcal{R}$. Each item v has its own KG, g_v, comprising an entity set $\mathcal{V}_v$ and a relation set $\mathcal{R}_v$ which contain features of v. We devise a set of item-entity alignments $\mathcal{A} = \{(v, e)|v \in \mathcal{I}, e \in \mathcal{V}\}$, where (v, e) indicates that item v is aligned with an entity e.

Given a user u and an item v represented by its KG g_v, the task is to generate an explanation of natural language sentences $E_{u,v}$ as to why item v was recommended for the user u. As in previous multi-task explainable recommendation models, KnowRec calculates a rating score $r_{u,v}$ that measures u's expected preference for v. By jointly training on the recommendation and explanation generation, our model can contextualize the embeddings more adequately with training signals from both tasks.

4 Model

Figure 1 illustrates our model with the user-item graph constructed through collaborative filtering signals, an encoder, and inference functions for explanation generation and rating prediction.

4.1 Input

The input of KnowRec comprises a user u represented by the user's purchase history $\{v_{ui}\}$ and an item v represented by its KG g_v, as introduced in Section 3. Let v_c denote the item currently considered by the system. The item v_c is aligned with one of the entities through $\mathcal{A}$ and becomes the center node of g_v, as shown in Figure 1.

Because our system leverages a Transformer-based encoder, we first linearize the input into a string. For the user $u = \{v_{ui}\}$, we initialize it by mapping each purchased item v_{ui} into tokens of the item's name. For the item v represented by g_v, we decompose g_v into a set of tuples $\{t_{vj}\}$, where $t_{vj} = (v_c, r_{vj}, n_{vj})$, $n_{vj} \in \mathcal{V}_v$, and $r_{vj} \in \mathcal{R}_v$. We linearize each tuple t_{vj} into a sequence of tokens using lexicalized names of the nodes and the relation. We then concatenate all the user tokens and the item tokens to form the full input sequence x. For example, suppose the current item v_c is the book *Harry Potter*, the KG has a single tuple (*Harry Potter*, *author*, *J.K. Rowling*), and the user previously purchased two books *The Lord of the Rings* and *The Little Prince*. In this case, input sequence x = *The Lord of the Rings The Little Prince Harry Potter author J.K. Rowling*.

We map the tokens to randomly initialized vectors or pre-trained word embeddings such as those in BART (Lewis et al., 2020), obtaining $\mathbf{X}_0 = [\ldots; \mathbf{V}_{ui}; \ldots; \mathbf{T}_{vj}; \ldots]$ where $\mathbf{V}_{ui}$ and $\mathbf{T}_{vj}$ are word vector representations of v_{ui} and t_{vj}, respectively. Unlike previous work on KG recommendation (Wang et al., 2020) where users/items are represented via purchase history and propagated KG information, our system infuses KG components to provide a recommendation and its natural language explanation. Our system also differs from prior studies on explainable recommendation in that while they focus on reviews and thus encode users/items as random vectors with additional review-based sparse token features as auxiliary information (Li et al., 2021), we directly encapsulate KG information into the input representation.

4.2 Encoder

Collaborative KG Representation. Because KnowRec outputs a natural language explanation grounded on KG facts, as well as a recommendation score for the user-item pair, we need to construct a user-item-linked KG to represent an input through its corresponding lexical graph feature. To do so, we leverage collaborative signals from $\mathbf{Y}$, combining u with v by linking previously purchased products v_{ui} to the current item v_c from g_v, forming a novel lexical user-item KG. Additionally, we connect all previously purchased items together in order to graphically model collaborative filtering effects for rating prediction, as illustrated in Figure 1. Note that the relations between previously purchased items and the current items do require a lexical representation for our model. The resulting graph goes through the Transformer architecture, as described below.

Global Attention. Transformer architectures have recently been adopted for the personalized explainable recommendation task (Li et al., 2021). We similarly leverage Transformer encoder layers (Vaswani et al., 2017), referred to as Global Attention, to encode the input representation with self-attention as:

$$\begin{aligned} &\mathbf{X}_l = \mathrm{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}, \\ &\mathbf{Q} = \mathbf{X}_{l-1}\mathbf{W}_l^Q, \mathbf{K} = \mathbf{X}_{l-1}\mathbf{W}_l^K, \\ &\mathbf{V} = \mathbf{X}_{l-1}\mathbf{W}_l^V \end{aligned} \tag{1}$$

where $\mathbf{X}_l$ is the output of the l-th layer in the encoder, and d_k is a tunable parameter. $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ represent the **Q**uery, **K**ey, and **V**alue vectors, respectively, each of which is calculated with the corresponding parameter matrix $\mathbf{W}$ in the l-th layer. Note that the transformer encoder may be initialized via a pre-trained language model.

User-Item Graph Attention. We further propose User-Item Graph Attention encoder layers, which compute graph-aware attention via a mask to capture the user-item graph's topological information, which runs in parallel with the Global Attention encoder layers.

We first extract the mask $\mathbf{M}_g \in \mathbb{R}^{m\times m}$ from the user-item linked KG, where m is the number of relevant KG components, i.e., nodes and edges that are lexically expressed in the KG (edges between v_{ui} and v_c not included). In $\mathbf{M}_g$, each row/column refers to a KG component. $M_{ij} = 0$ if there is a connection between component i and j (e.g., "J.K. Rowling" and "author") and $-\infty$ otherwise. In addition, we assume all item components, i.e., the previous purchases and the current item, are mutually connected when devising $\mathbf{M}_g$.

For each layer (referred to as the l-th layer), we then transfer its input $\mathbf{X}_{l-1}$ into a component-wise representation $\mathbf{X}^g_{l-1} \in \mathbb{R}^{m\times d}$, where d is the word embedding size. Motivated by Ke et al. (2021), we perform this transfer by employing a pooling layer that averages the vector representations of all the word tokens contained in the corresponding node/edge names per relevant KG component. With the transferred input $\mathbf{X}^g_{l-1}$, we proceed to encode it using User-Item Graph Attention with the graph-topology-sensitive mask as follows:

$$\begin{aligned} \tilde{\mathbf{X}}^g_l &= \mathrm{Attn}_M(\mathbf{Q}', \mathbf{K}', \mathbf{V}') \\ &= \mathrm{softmax}\left(\frac{\mathbf{Q}'\mathbf{K}'^\top}{\sqrt{d_k}} + \mathbf{M}_g\right)\mathbf{V}'. \end{aligned} \tag{2}$$

where query $\mathbf{Q}'$, key $\mathbf{K}'$, and value $\mathbf{V}'$ are computed with the transferred input and learnable parameters in the same manner as Equation (1).

Lastly, we combine the outputs of the Global Attention encoder and the User-Item Graph Attention encoder in each layer. As the two outputs have different dimensions, we first expand $\tilde{\mathbf{X}}^g_l$ to the same dimension of $\mathbf{X}_l$ through a *gather* operation, i.e., broadcasting each KG component-wise representation in $\tilde{\mathbf{X}}^g_l$ to every encompassing word of the corresponding component and connecting those representations. We then add the expanded $\tilde{\mathbf{X}}^g_l$ to $\mathbf{X}_l$ through element-wise addition, generating the

l-th encoding layer's output:

$$\tilde{\mathbf{X}}_l = gather(\tilde{\mathbf{X}}_l^g) + \mathbf{X}_l \tag{3}$$

Note, in this section, we illustrate the Global Attention encoder, User-Item Attention encoder, and their combination with single-head attention. In practice, we implement both encoders with multi-head attention as in Vaswani et al. (2017).

4.3 Rating Prediction

For the rating prediction task, we first separate and isolate user u and item v features via masking. Once isolated, we perform a mean pool on all their respective tokens and linearly project u and v to perform a dot-product between the two new vector representations as follows:

$$\begin{aligned} \tilde{\mathbf{x}}_u &= pool_{mean}(\tilde{\mathbf{X}}_L + \mathbf{m}_u)\mathbf{W}^u \\ \tilde{\mathbf{x}}_v &= pool_{mean}(\tilde{\mathbf{X}}_L + \mathbf{m}_v)\mathbf{W}^v \\ \hat{r}_{u,v} &= dot(\tilde{\mathbf{x}}_u, \tilde{\mathbf{x}}_v), \end{aligned} \tag{4}$$

where $\mathbf{m}_u$ and $\mathbf{m}_v$ are the user and item masks that denote which tokens belong to the user and item, $\mathbf{W}$s are learnable parameters, and L refers to the last layer of the encoder.

4.4 Explanation Generation

Before generating a final output text for our explanation, we pass the representation through a fully connected linear layer as the encoder hidden state and decode the representation into its respective output tokens through an auto-regressive decoder, following previous work (Lewis et al., 2020).

4.5 Joint-learning Objective

As previously noted, our system consists of two outputs: a rating prediction score $\hat{r}_{u,v}$ and natural language explanation $E_{u,v}$ which justifies the rating by verbalizing the item's corresponding KG. We thus perform multi-task learning to learn both tasks and manually define regularization weights λ, as in similar multi-task paradigms, to weight the two tasks. Taking $\mathcal{L}_r$ and $\mathcal{L}_e$ to represent the recommendation and explanation cost functions, respectively, the multi-task cost $\mathcal{L}$ then becomes:

$$\mathcal{L} = \lambda_r \mathcal{L}_r + \lambda_e \mathcal{L}_e, \tag{5}$$

where λ_r and λ_e denote the rating prediction and explanation regularization weights, respectively.

We define $\mathcal{L}_r$ using Mean Square Error (MSE) in line with conventional item recommendation and review-based explainable systems:

$$\mathcal{L}_r = \frac{1}{|\mathcal{U}||\mathcal{I}|} \sum_{u \in \mathcal{U} \wedge v \in \mathcal{I}} (r_{u,v} - \hat{r}_{u,v})^2, \tag{6}$$

where $r_{u,v}$ denotes the ground-true score.

Next, as in other NLG tasks (Lewis et al., 2020; Zhang et al., 2020), we incorporate Negative Log-Likelihood (NLL) as the explanation's cost function $\mathcal{L}_e$. Thus, we define $\mathcal{L}_e$ as:

$$\mathcal{L}_e = \frac{1}{|\mathcal{U}||\mathcal{I}|} \sum_{u \in \mathcal{U} \wedge v \in \mathcal{I}} \frac{1}{|E_{u,v}|} \sum_{t=1}^{|E_{u,v}|} -\log p_t^{e_t} \tag{7}$$

where $p_t^{e_t}$ is the probability of a decoded token e^t at time step t.

5 Dataset

Although KG-recommendation datasets exist, they do not contain any supervision signals to NL descriptions. Thus, to evaluate our explainable recommendation approach in a KG-aware setting and our KnowRec model, we introduce two new datasets based on the Amazon-Book and Amazon-Movie datasets (He and McAuley, 2016): (1) Book KG-Exp and (2) Movie KG-Exp.

Recall that our task requires an input KG along with an NL explanation and recommendation score. Because it is more efficient to extract KGs from text, rather than manually annotate each KG with text, we take a description-first approach, automatically extracting KG elements from the corresponding text. Given the currently available data, we leverage item descriptions as a proxy for the NL explanations, while constructing a user-item KG from an item's features and user's purchase history.

We first extract entities from a given item description via DBpedia Spotlight (Mendes et al., 2011), a tool that detects mentions of DBpedia (Auer et al., 2007) entities from NL text. We then query for each entity's most specific type and use those types as relations that connect the item to its corresponding entities. We construct a user KG via their purchase history, e.g. $[Purchase_1, Purchase_2, ...Purchase_n]$, as a complete graph where each purchase is connected. Finally, we connect all the nodes of the user KG to the item KG, treating each user purchase as a one-hop neighbor of the current item. To ensure the KG-explanation correspondence, we filter out any sentences in the explanation in which no entities were found. To measure objectivity, we calculate the proportion of a given KG's entities that appear in the explanation, called entity coverage (EC) (defined in Appendix B.2). We summarize our dataset statistics in Table 1 and present a more comprehensive comparison in Appendix A.2.

6 Experiments

6.1 Evaluation Metrics

We assess explainable recommendation following prior work: 1) on the recommendation performance and 2) on the explanation performance. For the explanation generation task, we employ standard natural language generation (NLG) metrics: BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004). We measure diversity and the detail-oriented features of the generated sentences using Unique Sentence Ratio (USR) (Li et al., 2020, 2021), and use EC, instead of feature coverage ratio, for coverage due to our non-review-based explanations.

6.2 Baseline Models

Previous models were primarily designed for user review data. To assess the effectiveness of our approach, we compare it to existing explanation generation baselines. These baselines include models that utilize user and item IDs, as well as those that employ word-level features. Additionally, we adapt several existing baselines to the context of explainable recommendation in a knowledge graph (KG) setting, addressing the need for adaptation, as the existing models were originally designed for user review data.

Att2Seq (Dong et al., 2017) was designed for review generation, where we adapt it to the item explanation setting. As in (Li et al., 2021), we remove the attention module, as it makes the generated content unreadable.

NRT (Li et al., 2017) is a multi-task model for rating prediction and tip generation, based on user and item IDs. As in previous work, we use our explanations as tips and remove the model's L2 regularizer (Li et al., 2020, 2021), which causes the model to generate identical sentences.

Transformer (Vaswani et al., 2017; Li et al., 2021) treats user and item IDs as words. We adapt the model first introduced for review generation by Li et al. (2021) while integrating the KG entities and relations instead of the review item features.

PETER (Li et al., 2021) utilizes both user/item IDs and corresponding item features extracted from user reviews to generate a recommendation score, explanation, and context related to the item features. The model also develops a novel PETER mask between item/user IDs and corresponding features/generated text. As our task does not take a feature-based approach, for a fair comparison we remove the context prediction module and input the whole KG into the model as the corresponding item features.

PEPLER (Li et al., 2022) is an extension of PETER, where the transformer is replaced with a pre-train language model, namely GPT-2 to generate both recommendation scores and explanations. We take the best-performing setting for a fair comparison, namely using the MLP setting for recommendation scores.

In addition to NRT, PETER, and PEPLER, as in previous work, we compare with two traditional baselines for recommendation: **PMF** (Mnih and Salakhutdinov, 2007) and **SVD++** (Koren, 2008).

6.3 Implementation

We train our newly proposed KnowRec model on two settings of the Book and Movie KG-Exp datasets, a full training set and a few-shot setting, where 1% of the data is used. Because our method provides item-level explanations based on KGs, we split the datasets based on their labeled description/explanation, and as such, we experiment in a setting where items in the test set can be unseen during training. By doing so, we handle a unique case that has not been considered in previous research relying on item reviews. The train/validation/test sets are split into 60/20/20. For KnowRec, we use BART as our pre-trained model, with a Byte-Pair Encoding (BPE) vocabulary (Radford et al., 2019). For more details regarding our experimental settings please see Appendix B.1.

7 Results and Analysis

7.1 Explanation Results

In Table 2, we evaluate the models' text reproduction performance using BLEU and ROUGE (R) metrics, while also examining their *explainability* through USR and EC analysis.

For BLEU and ROUGE, KnowRec consistently outperforms all baselines, achieving a BLEU-4 score of 10.71 and ROUGE-L F1 score of 27.71 on Movie KG-Exp and a BLEU-4 score of 12.60

Name	#Users	#Items	#Interactions	KG	#Es	#Rs	#Triples	EC	Desc.	Words/Sample
Book KG-Exp	*396,114*	*95,733*	*2,318,107*	*Yes*	*195,110*	*392*	*745,699*	*71.45*	*Yes*	*99.96*
Movie KG-Exp	*131,375*	*18,107*	*788,957*	*Yes*	*59,036*	*363*	*146,772*	*71.32*	*Yes*	*96.35*

Table 1: Statistics of our Book KG-Exp and Movie KG-Exp benchmark datasets. *#Es*, *#Rs*, and *Desc.* denote number of entities, number of relations, and if the dataset contains parallel descriptions.

Dataset	Model	BLEU-1	BLEU-4	USR	R2-F	R2-R	R2-P	RL-F	RL-R	RL-P	EC
Movie KG-Exp	Att2Seq	8.86	0.39	0.30	2.08	1.41	8.47	8.07	11.65	9.49	0.44
	NRT	11.76	0.57	0.03	1.50	1.40	3.25	7.20	11.70	8.05	0.98
	Transformer	8.67	0.18	0.33	1.21	0.91	6.55	6.58	9.54	9.69	0.82
	PETER	14.66	3.99	0.55	5.07	4.26	11.66	15.06	16.67	23.03	10.58
	PEPLER	11.68	0.13	0.46	0.56	0.63	0.54	8.90	10.92	9.53	0.78
	KnowRec	**37.02**	**10.71**	**0.83**	**15.49**	**15.12**	**18.15**	**27.71**	**28.71**	**37.10**	**67.97**
Book KG-Exp	Att2Seq	19.51	1.85	0.43	5.08	3.76	12.15	12.98	16.55	20.89	0.86
	NRT	21.06	2.59	0.10	6.18	4.88	11.44	15.57	18.67	24.36	1.57
	Transformer	16.90	2.01	0.12	5.68	4.23	11.94	13.66	15.57	26.87	2.08
	PETER	27.93	8.39	0.71	11.94	10.36	18.68	21.24	23.30	28.02	17.39
	PEPLER	16.07	1.20	0.90	2.39	2.63	2.26	13.03	16.34	12.24	0.74
	KnowRec	**38.53**	**12.60**	**0.92**	**19.78**	**19.44**	**23.22**	**28.29**	**29.43**	**35.28**	**69.50**

Table 2: Comparison of explanation generation models on the Movie KG-Exp and Book KG-Exp datasets.

and ROUGE-L F score of 28.29 on Movie KG-Exp. This suggests that previous baselines, designed for review-level explanation, are inadequate to produce longer and more objective explanations. Specifically, of the baselines, PETER which utilizes the whole lexical input, adapts best. However, KnowRec makes use of user-item graph encodings, which may lead to better generation of the item KG features mentioned in the ground truth texts. While PEPLER (Li et al., 2022)'s pretrained approach aids in fluent sentence generation, KnowRec excels in generating contextually relevant words around feature-level terms. Unlike PEPLER, which creates concise reviews based on user-item IDs, KnowRec utilizes graph attention to interconnect related components for comprehensive NL text explanations.

In terms of explainability, KnowRec also generates much more diverse sentences (USR), especially compared to models that do not leverage pre-trained models. Note that while PEPLER has a comparable USR score to KnowRec on the Book KG-Exp dataset, it does not similarly produce high-quality and related sentences according to the NLG metrics. Our results show that while the ground truth is based on item-level features, the generated output is still personalized as further discussed in Section 7.5. Also note the high discrepancy in EC, where the entity-level features are generated in the output text. As our goal is to generate objective and specific explanations, the EC can help real-world users understand what a certain recommended product is about and how it compares to other products. Therefore, it is crucial that explainable models capture these features when producing justifications for recommendations.

7.2 Few-shot Explanation Results

Real-world recommendation systems may face low-resource problems, where only a small amount of training data with few item descriptions is available but an item database exists. To reflect this practical situation, we also evaluate a few-shot setting where the training data is 1% of its total size.

As in previous experiments, we set the user-item size for KnowRec to 5. We show the results of this few-shot experiment in Table 3. KnowRec consistently and significantly outperforms other explainable baselines on both the Book and Movie datasets in terms of text quality, sentence diversity (USR), and entity representation (ER), showing our approach is effective even in data-scarce scenarios. Like KnowRec, PEPLER also leverages a pre-trained model, namely GPT-2. However, unlike KnowRec, the model does not adapt well to generating item-specific explanations. The second best model, PETER, fully leverages the KG features in their approach. However, such a model does produce diverse sentences. Note that those models that completely rely on user and item IDs, fail to produce quality explanations, as noted by their respective BLEU and ROUGE scores, showing the task to be more complex than previous explana-

Dataset	Model	BLEU-1	BLEU-4	USR	R2-F	R2-R	R2-P	RL-F	RL-R	RL-P	EC
Movie KG-Exp (Few-shot)	Att2Seq	2.63	0.00	0.00	0.00	0.00	0.00	2.73	4.25	2.63	0.01
	NRT	8.78	0.32	0.01	1.84	1.08	11.73	7.12	10.17	17.97	0.07
	Transformer	12.23	0.27	0.16	1.24	1.07	3.54	6.97	9.54	12.00	1.18
	PETER	12.28	0.68	0.36	2.33	1.45	12.49	12.00	13.18	18.03	5.44
	PEPLER	12.58	0.41	0.01	1.26	1.44	1.18	10.73	12.63	10.38	0.11
	KnowRec	**33.89**	**7.53**	**0.87**	**13.41**	**12.60**	**17.67**	**24.48**	**25.63**	**35.66**	**63.92**
Book KG-Exp (Few-shot)	Att2Seq	16.58	1.53	0.22	4.68	3.10	15.58	13.30	15.28	21.32	0.26
	NRT	19.12	2.19	0.01	6.11	4.36	13.99	15.18	20.47	16.78	1.19
	Transformer	12.69	1.22	0.08	3.60	3.16	8.65	9.77	15.64	10.58	1.57
	PETER	18.38	2.87	0.45	7.12	5.07	17.50	14.74	17.66	17.52	4.23
	PEPLER	7.96	0.26	0.02	0.67	0.63	0.83	7.59	10.07	7.04	0.54
	KnowRec	**28.93**	**7.94**	**0.93**	**17.28**	**16.05**	**22.45**	**24.84**	**25.19**	**36.60**	**60.46**

Table 3: Comparison of explanation generation models on the Movie KG-Exp and Book KG-Exp datasets in the few-shot learning setting (1% of training data).

Model	Book KG-Exp				Movie KG-Exp			
	All		Few		All		Few	
	R	M	R	M	R	M	R	M
PMF	3.50	3.35	3.50	3.35	3.31	3.08	3.32	3.08
SVD++	1.03	0.80	1.01	0.64	1.20	**0.79**	1.25	0.98
NRT	0.98	0.74	1.07	0.73	1.17	0.93	1.23	0.97
PETER	1.01	0.79	1.03	0.82	1.24	1.03	1.24	1.00
PEPLER	**0.96**	**0.72**	1.07	0.72	**1.14**	0.91	1.27	0.96
KnowRec	1.04	0.75	1.04	0.72	1.22	0.92	**1.21**	**0.93**

Table 4: Performance comparison on the recommendation task with respect to RMSE and MAE, denoted as R and M on the table respectively.

	BLEU-4↑	USR↑	RL-F↑	RMSE↓	MAE↓
KnowRec	7.94	**0.93**	24.84	**1.04**	**0.78**
- Recomm.	**8.32**	**0.93**	**24.90**	-	-
- UIG Att.	7.75	0.91	24.80	1.03	**0.78**

Table 5: Ablation study on the Book KG-Exp (Few-Shot) dataset. 'Recomm.' means the joint learning with recommendation scoring, and 'UIG Att.' denotes the user-item graph attention.

tion tasks relying on repetitive, short, and already existing user reviews.

7.3 Recommendation Performance

Table 4 shows the recommendation performance on all KG Explanation datasets. We report the Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) metrics to evaluate the recommendation task. As shown, all results except PMF are relatively close. PMF significantly underperforms due to the cold start problem presented on new items. KnowRec achieves performance comparable to other strong baselines, despite KnowRec being the only model that uses lexical features for the recommendation task, while the other models learn the task through user/item IDs. Thus, KnowRec may need more data to learn these parameters. Additionally, because we learn the recommendation task through lexical features, our model provides an interpretable solution that could be directly compared to the produced NL explanations.

7.4 Ablation Study

We perform ablation studies to analyze the effects of the recommendation and user-item graph components on Book KG-exp as shown in Table 5. Due to computational resources, we performed the study on the few-shot setting. We first examine the results of KnowRec without the recommendation module in the second row (- *Recomm.*). By removing the 'Recomm' component, the performance on the NLG metrics improves, as the task is now a single-objective generative task instead of a multi-objective. We next study the effects of the User-Item Attention encoders on KnowRec's explainability and recommendation performance (- *UIG Att*). As shown by - *UIG Att.*, even with a smaller training dataset of 1% of the full data, by removing this component, we observe a slight decrease in the NLG metrics, BLEU and ROUGE, and less diverse sentences (USR). The representation and attention masking on the user-item graph, which connects and encodes the local item information, may therefore give a better representation of the input which is in turn decoded to produce an explanation. This may be further expressed within larger datasets. Furthermore, from the NLG metric results, we can infer from Table 5 that our rating module does not significantly hinder the performance of the generation component of KnowRec.

7.5 Qualitative Analysis

To grasp KnowRec's effectiveness, we analyze explanations from Movie/Book KG-Exp test sets. These explanations are both grammatically smooth and adept at (1) integrating robust item features for factual insights and (2) tailoring personalized content based on diverse user purchase histories (examples in Appendix C, Table 7).

Consider the first two rows of the table, pertaining to the movie *Journey to the Center of the Earth*. We can see two different (but syntactically similar) generated explanations for two different users. In one case, the user has bought mystery and fantasy movies such as *Stitch in Crime*, *Columbo*, and *The Lord of the Rings*, and the output integrates related words such as *investigates* and *mysterious* to personalize the explanation. The second case mentions *classic* and *novel*, possibly because the second user's purchase history involves *Disney* classics and movies based on novels such as *The Hardy Boys* and *Old Yeller*. While the input KG does not explicitly state that *Journey to the Center of the Earth* is a novel, such information may be inferred from the KG's relation and supported through the user's related purchases. In both cases the output closely matches the ground truth, verbalizing item features from the KB such as *Jules Verne* and *magnetic storm*, suggesting that our model is robust in describing the explanation content, while still implicitly reflecting the user's purchase history.

8 Conclusion

We propose KnowRec, a Knowledge-aware model for generating NL explanations and recommendation scores on user-item pairs. To evaluate KnowRec, we devise and release a semi-supervised large-scale KG-NL recommendation dataset in the book and movie domain. Extensive experiments on both datasets demonstrate the suitability of our model compared to recently proposed explainable recommendation models. We hope that by proposing this KG-guided task, we will open up avenues to research focused on detailed, objective, and specific explanations which can also scale to new items and users, rather than the current review-focused work. In future work, we plan to incorporate user-specific KGs and other pre-trained language models into our model in order to verbalize both user and item-level feature explanations.

9 Limitations

While our approach generates objective, descriptive explanations while implicitly capturing personalized aspects of a user's purchase history, currently our dataset labels are limited to item-specific explanations, with the book-related KGs typically containing author-related information, and thus more information-dense than the movie-related KGs. These limitations are due to the currently available datasets, and future work can explore constructing a more personalized user-item KG for explainable recommendation. Furthermore, we represent users through their item purchase history in our approach. Therefore, while we handle the zero-purchase case for items (items that have not been purchased before), the zero-purchase case for users (users without a purchase history) is outside the scope of our work. In the future, we will extend our approach to user-attributed datasets to handle such cases.

10 Ethics Statement

All our experiments are performed over publicly available datasets. We do not use any identifiable information about crowd workers who provide annotations for these datasets. Neither do we perform any additional annotations or human evaluations in this work. We do not foresee any risks using KnowRec if the inputs to our model are designed as per our procedure. However, our models may exhibit unwanted biases that are inherent in pre-trained language models. This aspect is beyond the scope of the current work.

References

Nabiha Asghar. 2016. Yelp dataset challenge: Review rating prediction. *arXiv preprint arXiv:1605.05362*.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. DBpedia: A nucleus for a web of open data. In *Proceedings of the 6th International The Semantic Web and 2nd Asian Conference on Asian Semantic Web Conference*, pages 722–735, Berlin, Heidelberg. Springer-Verlag.

Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural attentional rating regression with review-level explanations. In *Proceedings of the 2018 World Wide Web Conference*, pages 1583–1592.

Hanxiong Chen, Shaoyun Shi, Yunqi Li, and Yongfeng Zhang. 2021. Neural collaborative reasoning. In *Pro-*

ceedings of the Web Conference 2021, pages 1516–1527.

Wenhu Chen, Yu Su, Xifeng Yan, and William Yang Wang. 2020. KGPT: Knowledge-grounded pre-training for data-to-text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8635–8648, Online. Association for Computational Linguistics.

Anthony Colas, Mehrdad Alvandipour, and Daisy Zhe Wang. 2022. GAP: A graph-aware language model framework for knowledge graph-to-text generation. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5755–5769, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Anthony Colas, Haodi Ma, Xuanli He, Yang Bai, and Daisy Zhe Wang. 2023. Can knowledge graphs simplify text? *arXiv preprint arXiv:2308.06975*.

Li Dong, Shaohan Huang, Furu Wei, Mirella Lapata, Ming Zhou, and Ke Xu. 2017. Learning to generate product reviews from attributes. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 623–632, Valencia, Spain. Association for Computational Linguistics.

Yuntao Du, Xinjun Zhu, Lu Chen, Baihua Zheng, and Yunjun Gao. 2022. HAKG: Hierarchy-aware knowledge gated network for recommendation. *arXiv preprint arXiv:2204.04959*.

Zuohui Fu, Yikun Xian, Ruoyuan Gao, Jieyu Zhao, Qiaoying Huang, Yingqiang Ge, Shuyuan Xu, Shijie Geng, Chirag Shah, Yongfeng Zhang, et al. 2020. Fairness-aware explainable recommendation over knowledge graphs. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 69–78.

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, Santiago de Compostela, Spain. Association for Computational Linguistics.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, volume 30.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th international conference on World Wide Web*, pages 507–517.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada. Association for Computational Linguistics.

Min Hou, Le Wu, Enhong Chen, Zhi Li, Vincent W Zheng, and Qi Liu. 2019. Explainable fashion recommendation: A semantic attribute region guided approach. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 4681–4688.

Bei Hui, Lizong Zhang, Xue Zhou, Xiao Wen, and Yuhui Nian. 2022. Personalized recommendation system based on knowledge embedding and historical behavior. *Applied Intelligence*, 52(1):954–966.

Pei Ke, Haozhe Ji, Yu Ran, Xin Cui, Liwei Wang, Linfeng Song, Xiaoyan Zhu, and Minlie Huang. 2021. JointGT: Graph-text joint representation learning for text generation from knowledge graphs. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2526–2538, Online. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434.

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, Online. Association for Computational Linguistics.

Chenliang Li, Cong Quan, Li Peng, Yunwei Qi, Yuming Deng, and Libing Wu. 2019. A capsule network for recommendation and explaining what you like and dislike. In *Proceedings of the 42nd International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 275–284.

Lei Li, Yongfeng Zhang, and Li Chen. 2020. Generate neural template explanations for recommendation.

In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 755–764.

Lei Li, Yongfeng Zhang, and Li Chen. 2021. Personalized transformer for explainable recommendation. 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 4947–4957, Online. Association for Computational Linguistics.

Lei Li, Yongfeng Zhang, and Li Chen. 2022. Personalized prompt learning for explainable recommendation. *arXiv preprint arXiv:2202.07371*.

Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. 2017. Neural rating regression with abstractive tips generation for recommendation. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 345–354.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Weizhi Ma, Min Zhang, Yue Cao, Woojeong Jin, Chenyang Wang, Yiqun Liu, Shaoping Ma, and Xiang Ren. 2019. Jointly learning explainable rules for recommendation with knowledge graph. In *The world wide web conference*, pages 1210–1221.

Pablo N Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. 2011. DBpedia spotlight: Shedding light on the web of documents. In *Proceedings of the 7th international conference on semantic systems*, pages 1–8.

Andriy Mnih and Russ R Salakhutdinov. 2007. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, volume 20.

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).

Leonardo F. R. 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, Online. Association for Computational Linguistics.

Shaoyun Shi, Hanxiong Chen, Weizhi Ma, Jiaxin Mao, Min Zhang, and Yongfeng Zhang. 2020. Neural logic reasoning. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1365–1374.

Peijie Sun, Le Wu, Kun Zhang, Yanjie Fu, Richang Hong, and Meng Wang. 2020. Dual learning for explainable recommendation: Towards unifying user preference prediction and review generation. In *Proceedings of The Web Conference 2020*, WWW '20, page 837–847, New York, NY, USA. Association for Computing Machinery.

Nava Tintarev and Judith Masthoff. 2015. Explaining recommendations: Design and evaluation. In *Recommender systems handbook*, pages 353–382. Springer.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *Proceedings of the International Conference on Learning Representations*.

Nan Wang, Hongning Wang, Yiling Jia, and Yue Yin. 2018a. Explainable recommendation via multi-task learning in opinionated text data. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 165–174.

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.

Xiang Wang, Tinglin Huang, Dingxian Wang, Yancheng Yuan, Zhenguang Liu, Xiangnan He, and Tat-Seng Chua. 2021. Learning intents behind interactions with knowledge graph for recommendation. In *Proceedings of the Web Conference 2021*, pages 878–887.

Xiting Wang, Yiru Chen, Jie Yang, Le Wu, Zhengtao Wu, and Xing Xie. 2018b. A reinforcement learning framework for explainable recommendation. In *2018 IEEE International Conference on Data Mining*, pages 587–596. IEEE.

Ze Wang, Guangyan Lin, Huobin Tan, Qinghong Chen, and Xiyang Liu. 2020. CKAN: Collaborative knowledge-aware attentive network for recommender systems. In *Proceedings of the 43rd International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 219–228.

Max Welling and Thomas N Kipf. 2016. Semi-supervised classification with graph convolutional networks. In *Proceedings of the International Conference on Learning Representations*.

Zhen Wu, Xin-Yu Dai, Cunyan Yin, Shujian Huang, and Jiajun Chen. 2018. Improving review representations with user attention and product attention for sentiment classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Yikun Xian, Zuohui Fu, Shan Muthukrishnan, Gerard De Melo, and Yongfeng Zhang. 2019. Reinforcement knowledge graph reasoning for explainable recommendation. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, pages 285–294.

Lijie Xie, Zhaoming Hu, Xingjuan Cai, Wensheng Zhang, and Jinjun Chen. 2021. Explainable recommendation based on knowledge graph and multi-objective optimization. *Complex & Intelligent Systems*, 7(3):1241–1252.

Aobo Yang, Nan Wang, Hongbo Deng, and Hongning Wang. 2021. Explanation as a defense of recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 1029–1037.

Wenhao Yu, Chenguang Zhu, Zaitang Li, Zhiting Hu, Qingyun Wang, Heng Ji, and Meng Jiang. 2022. A survey of knowledge-enhanced text generation. *ACM Computing Surveys*, 54(11s):1–38.

Yongfeng Zhang, Xu Chen, et al. 2020. Explainable recommendation: A survey and new perspectives. *Foundations and Trends® in Information Retrieval*, 14(1):1–101.

Yaxin Zhu, Yikun Xian, Zuohui Fu, Gerard de Melo, and Yongfeng Zhang. 2021. Faithfully explainable recommendation via neural logic reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3083–3090, Online. Association for Computational Linguistics.

A Dataset Details

A.1 Source Data

Amazon product data: The Amazon product dataset is a large-scale widely used dataset for product recommendation containing product reviews and metadata from Amazon. Data fields include ratings, texts, descriptions, and category information (He and McAuley, 2016). Because the dataset contains item descriptions, we can leverage such data to extract entities and relations to construct a KG that matches the textual description. Thus, these descriptions provide objective, item-distinct explanations as to why a user may have purchased a product. Although a user may not have reviewed an item, the dataset provides an existing description of the item, allowing models to produce explanations for such items. To keep our datasets large-scale, we focus on Amazon Book and Amazon Movie 5-core, the two largest Amazon product datasets.

A.2 Dataset Comparison

Table 6 summarizes existing popular recommendation system datasets utilized for both the explainable recommendation and KG recommendation task. We report both traditional recommendation features, KG-recommendation features, and explainable recommendation features. Last.FM (Wang et al., 2019), Book-Crossing (Wang et al., 2020), MovieLens20M (Wang et al., 2020), and Amazon-book (KG) (Wang et al., 2019) are popular benchmarks for the KG-recommendation task but contain no NL explanation features. Yelp-Restaurant, Amazon Movies & TV, and TripAdvisor-Hotel have been recently experimented with for the explainable recommendation task (Li et al., 2020), but lack KG data and rely on user reviews as proxies for the explanation. In contrast, our datasets, referred to as *Book KG-Exp* and *Movie KG-Exp* contain both KG and the corresponding parallel item descriptions associated with those KGs as explanations. Compared to Book KG-Exp, the Movie KG-Exp dataset contains fewer amount of unique KG elements, with *59,036* to *195,110* and *745,699* to *146,772* unique entities and KG, while having similarly sized explanations.

A.3 Dataset Statistics

We provide detailed statistics on both the Book KG-Exp and Movie KG-Exp datasets in Figure 2. As seen in Figures 2(a) and 2(b), the distributions of KGs with respect to the number of tuples shows similar long-tail distributions in both datasets. We observe from Figures 2(c) and 2(d) that a similar trend of long-tail distributions exists for both with respect to explanation lengths, where the lengths in the book dataset tend to skew more right than the lengths in the movie dataset.

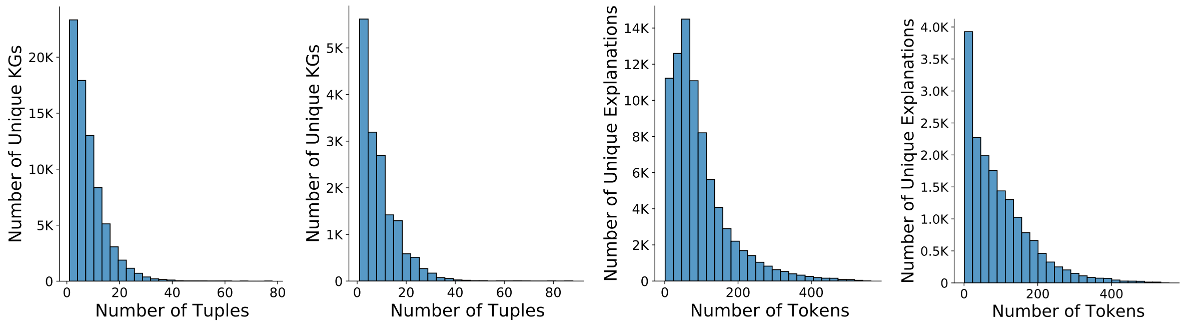
B Experiment Details

B.1 Hyper-parameters and Settings

As in (Li et al., 2021), we adapt the baseline codes to our setting and set the vocabulary size for NRT,

Name	#Users	#Items	#Interactions	KG	#Es	#Rs	#Triples	Desc.	Words/ Sample
Last.FM	23,566	48,123	3,034,796	Yes	58,266	9	464,567	No	-
Book-Crossing	276,271	271,379	1,048,575	Yes	25,787	18	60,787	No	-
Movie-Lens20M	138,159	16,954	13,501,622	Yes	102,569	32	499,474	No	-
Amazon-book (KG)	70,679	24,915	847,733	Yes	88,572	39	2,557,746	No	-
Yelp-Restaurant	27,147	20,266	1,293,247	No	-	-	-	No	12.32
Amazon Movies	7,506	7,360	441,783	No	-	-	-	No	14.14
TripAdvisor-Hotel	9,765	6,280	320,023	No	-	-	-	No	13.01
Book KG-Exp	*396,114*	*95,733*	*2,318,107*	*Yes*	*195,110*	*392*	*745,699*	*Yes*	*99.96*
Movie KG-Exp	*131,375*	*18,107*	*788,957*	*Yes*	*59,036*	*363*	*146,772*	*Yes*	*96.35*

Table 6: Comparison of widely used datasets divided by task: KG-Recommendation (top), Explainable Recommendation (middle), and KG Explainable Recommendation (bottom).



(a) Book KGs (b) Movie KGs (c) Book Explanations (d) Movie Explanations

Figure 2: Distributions for number of tuples (Figures 2(a) and 2(b)) and tokens (Figures 2(c) and 2(d)) per sample.

ATT2Seq, and PETER to 20,000 by keeping the most frequent words. For PETER and PEPLER, we set the number of context words to 128. For all approaches, including KnowRec, we set the length of explanation to 128, as the mean length is about 94 for both datasets. For KnowRec, we use an embedding size of 512, using a Byte-Pair Encoding (BPE) vocabulary (Radford et al., 2019) of size 50,256, with 2 encoding layers. Following KG generation work (Ribeiro et al., 2021), we split the tokens in the linearized graph with their corresponding label: *[user], [graph], [head], [relation], and [tail]*. For both datasets, we set the batch size to 128 and max user and KG size to 64 and 192, respectively. We set the max node and edge length to 60. We experiment with λ_r and λ_e and find that 0.01 and 1 give us the best BLEU performance without affecting the recommendation prediction scores as in (Li et al., 2022). See Figure 3 for an analysis of Movie KG-Exp (Few-shot). The model's parameters were trained for 20 epochs and optimized via Adam (Kingma and Ba, 2015) with a learning rate of 1e-3 and *Adam* ϵ of 1e-08, and the gradients were clipped at 1.0. All other attention-related hyper-parameters were the same as used in previous work (Lewis et al., 2020). We decoded the text via beam search (Hokamp and Liu, 2017) with a beam size of 5. Experiments were performed on NVIDIA RTX 3090 GPUs. We evaluate the model based on the validation set's total loss instead of BLEU score due to computational limitations, saving the top 10 models for testing, because the model with the least loss does not necessarily result in the best NLG metrics.

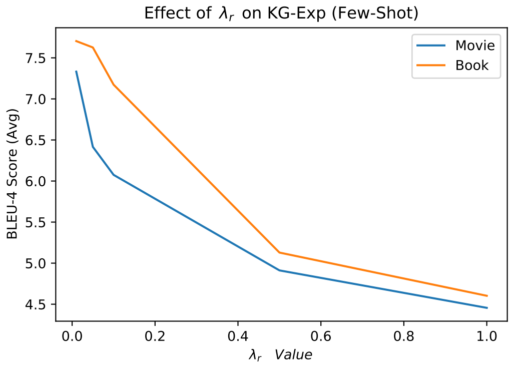


Figure 3: Effect of λ_r on the BLEU-4 score for the Book and Movie KG-Exp datasets. We average all top 10 runs for a more comprehensive comparison.

Because of computation limitations, for evaluation purposes, we randomly sample and evaluate on 1% of the test set, containing 4,491 and 1,456 samples for the Book and Movie datasets respectively. Note, that the size of the test set is comparative to other text generative tasks such as KG-to-text (Gardent et al., 2017) and summarization (Yu et al., 2022).

B.2 Entity Coverage

We define entity coverage (EC) as the percentage of unique entities, originating in an item KG, which appears in the recommendation explanation. More formally, for each head and tail entity *e* in an item KG's set of entities *E*, we calculate the token overlap in the explanation output for those entities. The EC score ranges in $[0, 1]$, where we report the percentage value in our results. The Book KG-Exp and Movie KG-Exp had an EC score of 71.45% and 71.32%, indicating that a descriptive, objective explanation should have a high EC score. The formula for EC is defined as:

$$\frac{\#KG\, entities\, found\, in\, output}{\#KG\, entities}$$

or is the recall of the entities in a KG.

C Generated Examples

Table 7 presents some examples generated by KnowRec from the Book and Movie KG-EXP datasets. As discussed in Section 7, we find the examples to be fluent and grammatical, while incorporating both item features and implicit user information based on a user's purchase history. The generated examples closely match the ground truth, while integrating some language derived from the user. Note, that our aim here is to illustrate examples that showcase the implicit user preferences, instead of showing those generated outputs which most closely match the ground truth descriptions. As with other state-of-the-art NLG models, KnowRec does have a tendency to hallucinate by adding extra information that may not be necessarily accurate. As can be by the NLG metrics in Table 2, KnowRec relieves the hallucination problem by incorporating the user-item KG information. Such limitations may be additionally improved by leveraging more dense background KGs to generate from, while also incorporating user purchase history item features.

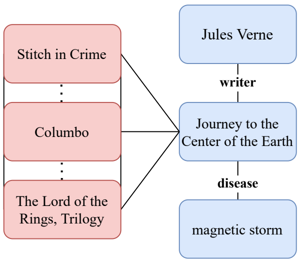
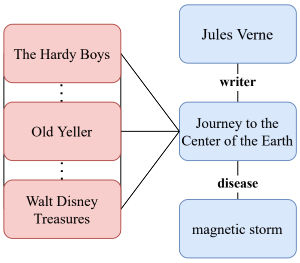
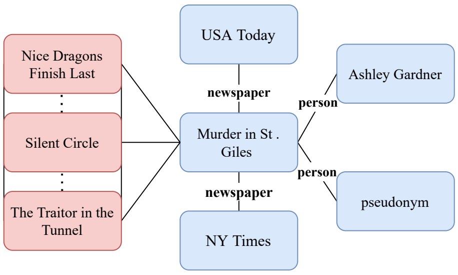
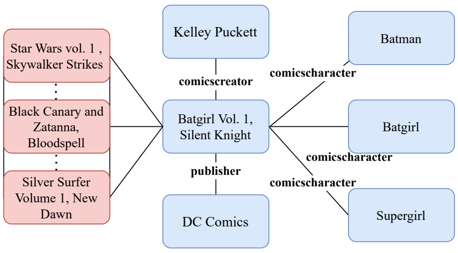
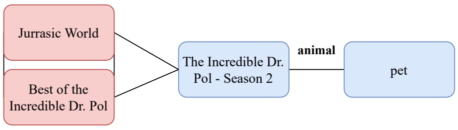
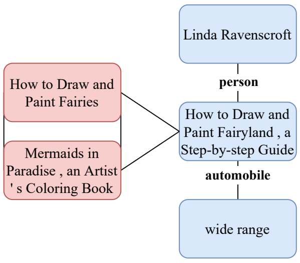
Item Graph Representation	Generated Explanation	Ground Truth Explanation
 Stitch in Crime Columbo The Lord of the Rings, Trilogy Jules Verne writer Journey to the Center of the Earth disease magnetic storm	a scientist (**jules verne**) investigates a **magnetic storm** that sends a mysterious beam of light from earth **to the center of earth**.	jules verne's professor lindenbrook leads a trip through monsters, mushrooms and a magnetic storm.
 The Hardy Boys Old Yeller Walt Disney Treasures Jules Verne writer Journey to the Center of the Earth disease magnetic storm	a group of scientists, inspired by **jules verne's** classic novel, take a trip to the **magnetic storm** at **the center of the earth**.	jules verne's professor lindenbrook leads a trip through monsters, mushrooms and a magnetic storm.
 Nice Dragons Finish Last Silent Circle The Traitor in the Tunnel USA Today newspaper Murder in St . Giles newspaper NY Times person Ashley Gardner person pseudonym	**ashley gardner** is a **ny times** and **usa today** bestselling author. under the **pseudonym** jennifer ashley, she has collectively written more than 70 mystery and historical novels.	usa today bestselling author ashley gardner is pseudonym for ny times bestselling author jennifer ashley.
 Star Wars vol. 1 , Skywalker Strikes Black Canary and Zatanna, Bloodspell Silver Surfer Volume 1, New Dawn Kelley Puckett comicscreator Batgirl Vol. 1, Silent Knight publisher DC Comics comicscharacter Batman Batgirl comicscharacter comicscharacter Supergirl	**kelley puckett** is an american comic book writer best known for his work on **batman** for **dc comics**. he is the author of numerous books for young readers, including **supergirl**, the ultimate guide to character development and **batgirl**, a guide to writing for comics, both published by image.	kelley puckett has been writing comics for far too long, by general consensus. he has worked on such series as batman adventures, batgirl and kinetic and supergirl for dc comics.
 Jurrasic World Best of the Incredible Dr. Pol The Incredible Dr. Pol - Season 2 animal pet	your favorite **dr. pol** vet and his **pet** dog return for a second season of this hilarious and heartwarming animated adventure.	from sick goats to sick pet pigs, dr. pol and his colleagues have their hands full with a variety of cases and several animal emergencies.
 How to Draw and Paint Fairies Mermaids in Paradise , an Artist ' s Coloring Book Linda Ravenscroft person How to Draw and Paint Fairyland , a Step-by-step Guide automobile wide range	**linda ravenscroft** is an award-winning children's book author and illustrator who has illustrated a **wide range** of books and magazines, including the best-selling how to draw and paint series.	linda ravenscroft has produced a wide range of images in fairyland motifs, including fine art prints, exclusive giftware, and fantasy art books.

Table 7: Examples generated by KnowRec on the Book/Movie KG-Exp datasets. In the first column, we follow the format of user-item KG representation in Figure 1, where red nodes represent a user's purchase history and blue nodes represent an item KG. For clarity and brevity, we only show the relevant parts of the item graphs. In the second column, the bold words are the item features directly coming from the item KG representation, whereas the underlined words are the features implicitly captured by KnowRec, based on the user's purchase history.