A Model-Agnostic Recommendation Explanation System Based on Knowledge Graph

Yuhao Chen and Jun Miyazaki(✉)

Tokyo Institute of Technology, 2-12-1 Ookayama, Meguro-ku, Tokyo, Japan
chen@lsc.c.titech.ac.jp, miyazaki@c.titech.ac.jp

Abstract. Recommender systems have been gaining attention in recent decades for the ability to ease information overload. One of the main areas of concern is the explainability of recommender systems. In this paper, we propose a model-agnostic recommendation explanation system, which can improve the explainability of existing recommender systems. In the proposed system, a task-specialized knowledge graph is introduced, and the explanation is generated based on the paths between the recommended item and the user's history of interacted items. Finally, we implemented the proposed system using Wikidata and the MovieLens dataset. Through several case studies, we show that our system can provide more convincing and diverse personalized explanations for recommended items compared with existing systems.

Keywords: Recommender system · Knowledge graph · Model-agnostic · Explainability · Justification

1 Introduction

Due to the scale of the Internet and the rapid growth of information resources, it is becoming difficult for people to obtain desired information from a large amount of data. Recommender systems play an important role in all aspects of our daily lives as they are one of the main methods for addressing this information overload. In recent decades, recommender systems have been increasingly researched and significant progress has been made. High-quality personalized explanations of recommendations can boost trust, effectiveness, efficiency, and satisfaction [13]. As such, the explainability of recommender systems is one of the main areas of concern. Since the widely used machine learning-based recommender systems are lacking in explainability, the explainability of recommender systems has become more important than ever.

Explainability serves two purposes in recommender systems: transparency and justification.

– Transparency is the property that clarifies the recommendation process and enables users to understand the mechanism of the system.

S. Hartmann et al. (Eds.): DEXA 2020, LNCS 12392, pp. 149–163, 2020.
https://doi.org/10.1007/978-3-030-59051-2_10

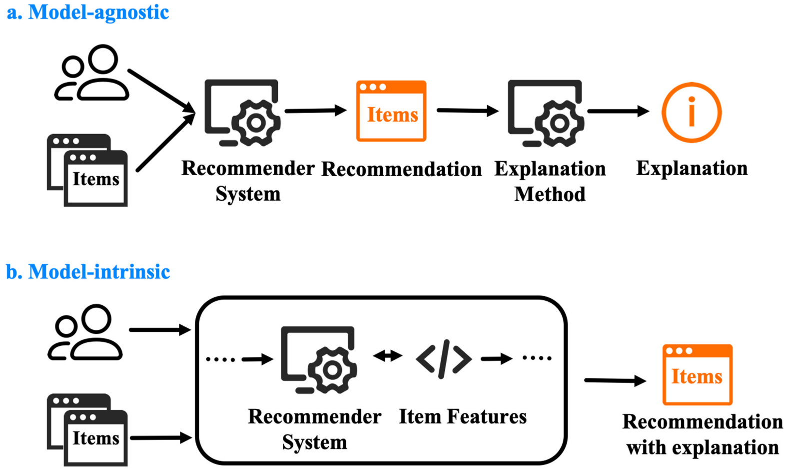

Fig. 1. Model-agnostic approach and Model-intrinsic approach.

– Justification is the property that provides consistent explanations independent of the recommendation module.

These two purposes correspond to two different implementations, model-agnostic and model-intrinsic approaches. The model-intrinsic method integrates the explanation process into a recommendation model, as shown in Fig. 1b. Therefore, its recommender mechanism is transparent on the some level [4]. The model-agnostic approach treats a trained recommendation module as a black-box and uses an independent method to explain the recommended item, as shown in Fig. 1a. The latter approach is particularly important for modern recommender systems because it can provide reasons for recommended items even when the recommendation module is too complex to be interpreted by uses. Specifically, in industries, most recommender systems are based on very complicated hybrid models which make it almost impossible to use model-intrinsic model to generate explanation. The model-agnostic model can also provide explainability when the system provider does not want to disclose the algorithm in the recommendation module [18]. The explainability discussed in this work is for justification purposes.

Since the model-agnostic model normally uses only information on users and items, as shown in Fig. 1a, variety of explanations is limited and the explanations lack persuasiveness. To address the aforementioned problems, we introduce a task-specialized knowledge graph to the model-agnostic approach, as shown in Fig. 2. The underlying assumption is that the information contained in datasets is not enough to generate high-quality personalized explanations. Therefore, additional *General common knowledge* is needed. In this paper, a knowledge graph is used as General common knowledge for generating high-quality personalized explanations.

Fig. 2. Proposed approach

2 Related Work

2.1 Recommender System

A recommender system extracts useful information for users and predicts users' preferences on their unseen items. Recommender systems are classified into collaborative filtering-based, content-based, and a hybrid of the two [1]. In particular, collaborative filtering recommender systems have been an important research direction for a long time. These systems make recommendations by capturing user-item relationships. Classic collaborative filtering algorithms, such as ItemKNN [15] and SAR [2], are all important milestones for information retrieval. Recently, deep-learning-based recommender systems, such as NCF [9], Wide and Deep Model [7], CNN+attention [8], have gained significant attention due to their ability to capture non-linear and non-trivial user-item relationships.

2.2 Knowledge Graphs-Based Explanation

Let E be a set of entities and R a set of edges labeled with relations. A knowledge graph (KG) is defined as a set of triples as follows:

$$KG = \{(h, r, t)|h, t \in E, r \in R\}, \tag{1}$$

where a triple (h, r, t) indicates that the head entity h and the tail entity t have a relation r.

Due to the volume of information contained in a knowledge graph, it can help to generate intuitive and more personalized explanations for recommended items. Knowledge graph-based explainable recommender systems have been explored in [3,6,10,14]. For example, Wang et al. [14] proposed an end-to-end explainable recommender system by simulating the ripples propagating on the surface of the water on the knowledge graph. Ma et al. [10] proposed a joint learning model that integrates explainable rule induction in a knowledge graph with a rule-guided neural recommendation model to leveraged machine learning and rule learning.

However, the models in these studies were model-intrinsic. They cannot be applied directly to an efficient and stable existing black-box recommendation model.

2.3 Model-Agnostic Based Explanation

Due to its ability to provide explanations for complex black-box recommender systems without affecting underlying recommendation algorithms, the model-agnostic based approach has been the main research direction for explainable recommender systems [11,12,17]. For example, Peake et al. [11] proposed a model-agnostic explainable recommender system based on association rule mining. The recommendation model in the paper is treated as a black-box, because it is based on a matrix factorization method. For each user, the user history and the recommended items constitute a transaction, and the association rules are extracted from the transactions of all users. When the same recommended items can be generated from these association rules, the author uses association rules as explanations of recommended items. Singh et al. [12] proposed another perspective of constructing the model-agnostic recommender system based on learning-to-rank algorithms. A black-box ranker is trained first, and then, an explainable tree-based model is trained with the ranking labels produced by the ranker. Finally, the model-agnostic explanations are generated by the tree-based model.

Despite their efforts to provide high-quality personalized explanations, these studies have largely been unable to provide varying persuasive explanations when facing sparse user-items datasets.

3 Proposed System

As shown in Fig. 3, the proposed system consists of two parts, the recommendation module and the explanation module. The recommendation module generates the items to be recommended based on the underlying recommender system. The recommender system is trained by the user-item information and outputs recommended items based on the predicted user preferences. The explanation module takes the recommended item as input and outputs the explanation of why the item was recommended.

The proposed system is model-agnostic; the recommendation module can utilize any mainstream recommender system that generates recommended items based on the user's item interaction history. Therefore, we focus on the explanation mechanisms in our proposed system.

The explanation module generates explanations of recommended items in the following two steps:

1. Item Knowledge Graph Generation.
2. Explanation Generation.

3.1 Item Knowledge Graph Generation

A general-purpose open knowledge graph is used in the proposed system. Existing general-purpose open knowledge graphs, such as Wikidata, DBpedia, etc., are

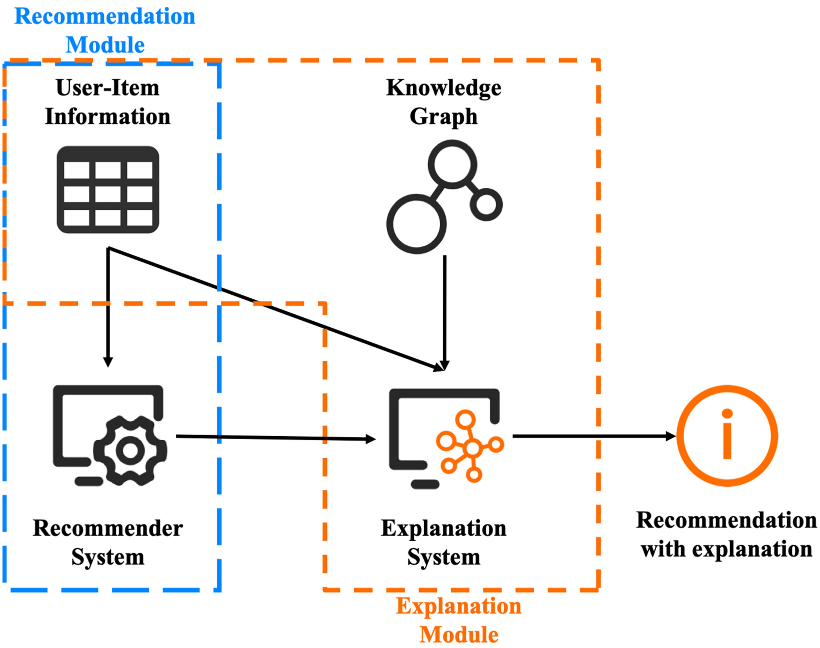

Fig. 3. Proposed system

enormous and have more than 100 million connections that contain noise. Therefore, they are not suitable for directly generating explanations of recommended items. In our method, the relevant parts of a knowledge graph are extracted from a general-purpose open knowledge graph in order to attain high-quality personalized explanations. The relevant portion of the knowledge graph is referred to as the *Item Knowledge Graph (Item KG)* in this paper.

We define the Item KG here. The procedure for generating the Item KG is shown on the left in Fig. 4.

First, a Domain Knowledge Graph (Domain KG), KG_D, is extracted from a general-purpose open knowledge graph, which is represented by

$$KG_D = \{(h, r, t) | h, t \in E, r \in R_D\}, \tag{2}$$

where R_D refers to the relations related to the recommended task. Note that the method of choosing relations is not unique because explanations can be applied in a variety of scenarios, even for the same recommendation task.

After extracting a set of entities E_I and a set of relations R_I in user-item information, we map E_I and add R_I to KG_D. All triples unrelated to the items in the user-item information are removed from KG_D

Finally, an Item KG KG_I is constructed with KD_D as follows:

$$KG_I = \{(h, r, t) | h, t \in E \cup E_I, r \in R_D \cup R_I\}. \tag{3}$$

The structure of the Item KG is shown in Fig. 5.

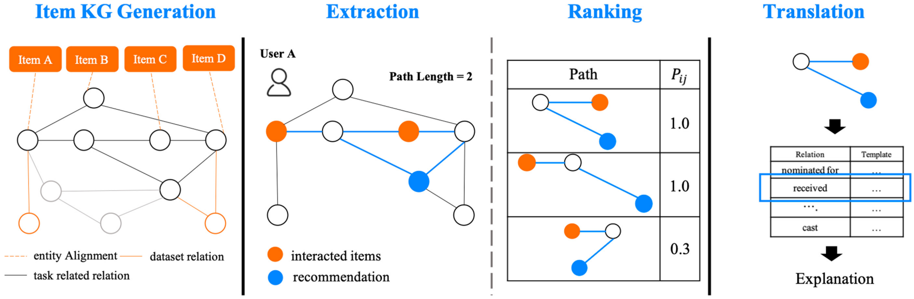

Fig. 4. Explanation module (Color figure online)

3.2 Explanation Generation

The explanations are generated from the Item KG through extraction, ranking, and translation processes.

The extraction process extracts the candidate paths on the Item KG based on the item recommended by the recommendation module and the user's interaction history. As shown in Fig. 4 (center), the target items in the user's interaction history (the orange nodes in Fig. 4) and the recommended item (the blue node in Fig. 4) are respectively set as start points and an end point. All of the paths with a length of d or less (blue paths in Fig. 4) are extracted from the Item KG as the target user's candidate paths.

Typically, there are multiple paths between the item recommended to a user and his/her history of interacted items. Since it is impractical to use all the paths for generating explanations for only one recommended item, a ranking process is needed in order to choose the most relevant paths for generating effective personalized explanations. The ranking process is based on the user's preference for the entities in candidate paths.

Let $P_{i,j}$ be user i's preference for the entity E_j. $P_{i,j}$ is calculated by the following equation:

$$P_{i,j} = \frac{|E_j^*|}{|E_j|}, \tag{4}$$

where $|E_j|$ is the total number of items directly connected to E_j (except for the recommended item), and $|E_j^*|$ is the total number of E_j directly connected to the items with which user i has interacted (except for the recommended item). For example, user A's preferences for the entities shown in Fig. 4 (center) are 1.0 and 0.3. The ranking process selects the top k entities based on $P_{i,j}$. Note that d and k are tuning parameters, which can be changed based on the task and the properties of the Item KG. In general, the persuasiveness of the path decreases as the path length becomes longer. For every selected entity, the proposed system randomly chooses one of the candidate paths as an explanation path.

Finally, the translation process translates the explanation path into a natural language. This is implemented by using templates prepared in advance. A variety

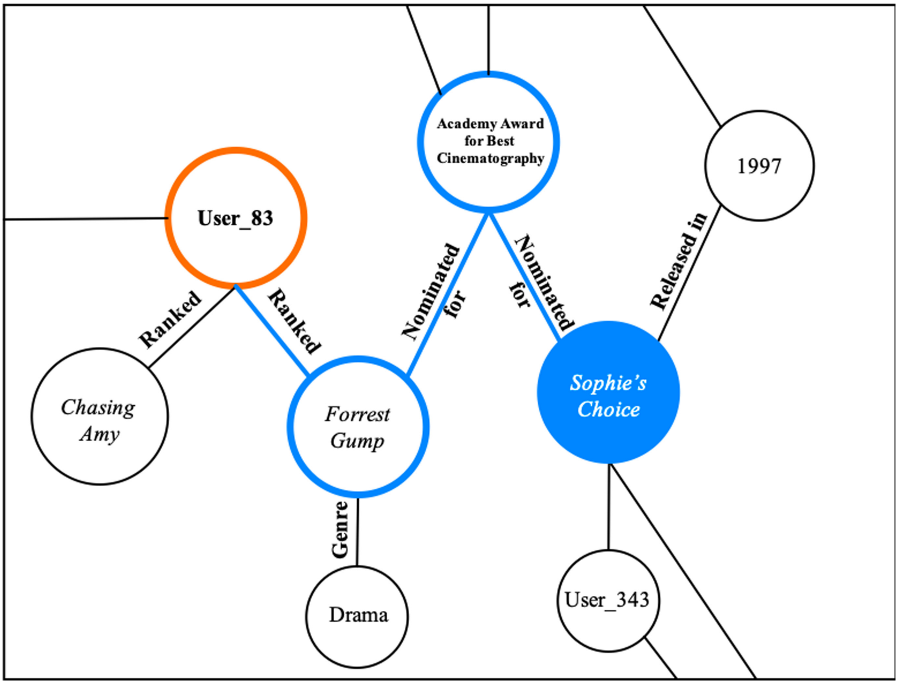

Fig. 5. Example of Item Knowledge Graph

of templates are created for each relation in the Item KG, in contrast to the conventional approaches which only use a single template.

An example based on the Item KG is shown in Fig. 5. Suppose that the item recommended to User_83 is movie *Sophie's Choice*, and the path selected is

> User_83 -Ranked- *Forrest Gump* -Nominated for - Academy Award for Best Cinematography -Nominated for- *Sophie's Choice*

Using the template "The movie __ was also nominated for __, like the movie __ you viewed before"for the relations Ranked and Nominated for, the explanation of this recommendation can be generated as:

> The movie *Sophie's Choice* was also nominated for the Academy Award for Best Cinematography, like the movie *Forrest Gump* which you viewed before.

Moreover, this approach can be extended to the case in which no candidate path exists between the target items in the user's interaction history and recommended items. In such a case, an unpersonalized explanation can be generated based on the popularity of the entities related to a recommended item. Now, we define popularity Pop_{ij} for item i's related entity $E_j \in \{t|(E_i, r, t) \in KG_I\}$. Pop_{ij} is calculated by

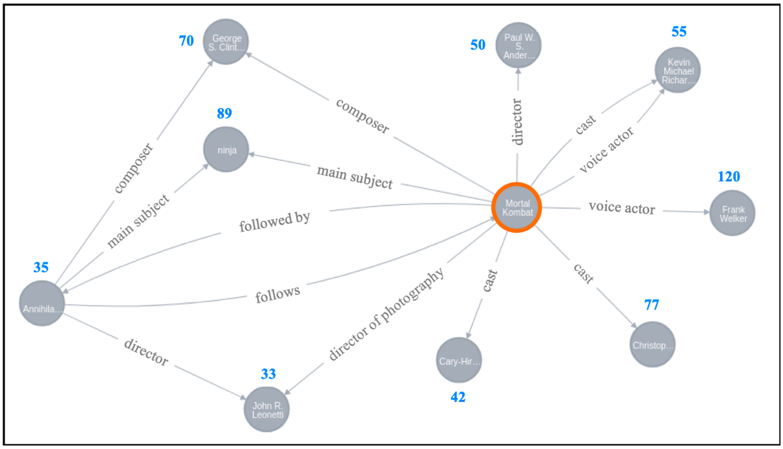

Fig. 6. Example of un-personalized explanation

$$Pop_{ij} = deg(E_j), \tag{5}$$

where $deg(E_j)$ is the degree of node j in KG_I.

For example, when the recommended movie *Mortal Kombat* and the popularity of its related entities are shown in Fig. 6, entities with high popularity such as *ninja* can be chosen. Then, we use a template to generate an unpersonalized explanation, such as:

How about *Mortal Kombat* whose subject is *ninja*?

4 Experiment

4.1 Dataset

We used the MovieLens[1] dataset as user interaction history data to train recommendation models and constructed an Item KG. In order to evaluate the proposed model more comprehensively, three different sized MovieLens datasets (MovieLens-100k, MovieLens-1m, MovieLens-20m) are introduced in this experiment. Table 1 shows the detail of MovieLens datasets used in this experiment.

To construct the knowledge graph, we used Wikidata[2] as the basis of the Item KG. We extracted a Movie-related subset KG from the WikiData dataset because the original Wikidata dataset was too large to be processed. The movie-related datasets were extracted from the Wikidata archive dump[3] by using python package WiKiDataSet [5]. The following three steps are performed to extract movie-related entities and relations from the Wikidata archive dump.

[1] https://grouplens.org/datasets/MovieLens/.
[2] https://www.wikidata.org/wiki/Wikidata.
[3] https://dumps.wikimedia.org/wikidatawiki/entities/.

Table 1. Details of MovieLens dataset

	# users	# movies	# ratings	# genres
MovieLens-100k	943	1,682	100,000	19
MovieLens-1m	6,040	3,900	1,000,209	19
MovieLens-20m	138,493	27,278	20,000,263	19

1. Get the Wikidata entities which are sub-classes of the film topic.
2. Find the lines corresponding to the selected entities in the Wikidata archive dump.
3. Organize the data collected in Step 2 into a triplet format.

4.2 Knowledge Graph

Table 2. Knowledge graph statistics

	# entities	# relations	# triples
Movie-related Dataset	518,175	280	3,113,902
Slack Domain Knowledge Graph	477,462	97	1,865,064
Strict Domain Knowledge Graph	451,545	52	1,580,100
Slack Item Knowledge Graph 100k	30,163	65	52,719
Slack Item Knowledge Graph 1m	43,506	68	103,344
Slack Item Knowledge Graph 20m	115,879	88	438,853
Strict Item Knowledge Graph 100k	28,189	43	43,298
Strict Item Knowledge Graph 1m	40,426	45	83,490
Strict Item Knowledge Graph 20m	107,202	50	351,553

Two sets of relations are manually chosen from the movie-related dataset to construct the Domain KG, strict Domain KG, and slack Domain KG. The slack Domain KG contains 97 kinds of relations. We excluded the relations unsuitable for generating explanations of recommendations from the movie-related dataset, such as "box office" and "Australian Classification". The strict Domain KG contains 52 relations. We also excluded the relations that cannot generate a persuasiveness explanation in the experiments on the slack Domain KG. The two relation datasets are shown in Fig. 7 and Fig. 8. The statistics of the Domain KGs can be found in Table 2.

For each slack and strict Domain KG, three Item KGs were constructed based on MovieLens-100k, MovieLens-1m, and MovieLens-20m. We prepared a total of six types of Item KGs for the experiment.

Relation	Relation	Relation	Relation	Relation	Relation	Relation
director of photo	based on	named after	inspired by	has list	time period	assistant director
director	main subject	main category	archives at	list of characters	member of	subclass of
screenwriter	nominated for	part of the series	creator	lyrics by	heritage design	copyright status
production com	followed by	fabrication method	sport	series spin-off	conflict	influenced by
cast member	different from	uses	distribution	cites	scenographer	opposite of
producer	movement	place in fictional	publisher	country	participant	item operated
country of origin	production design	voice actor	conferred by	editor	location	sponsor
narrative location	performer	part of	dedicated to	librettist	recorded at	plot expanded in
filming location	costume designer	derivative work	make-up artist	fictional universe	place publication	presented in
film editor	follows	characters	narrator	contributors	owned by	manifestation of
composer	collection	author	storyboard artist	sound designer	participant of	film script
distributor	soundtrack album	depicts	art director	competition class	location	commemorates
award received	after a work by	set in period	musical conductor	educated at	operator	
has quality	executive producer	theme music	choreographer	catalog	produced by	

Fig. 7. Slack relationship set

Relation	Relation	Relation	Relation	Relation	Relation	Relation
director of photo	based on	named after	inspired by			
director	main subject	main category				
screenwriter	nominated for	part of the series	creator			
	followed by			series spin-off	conflict	influenced by
cast member		uses		cites	scenographer	
		place in fictional				
country of origin	production design	voice actor	conferred by	editor		
		part of	dedicated to	librettist		
		derivative work		fictional universe		
film editor	follows	characters	narrator	contributors		
composer		author				film script
		depicts	art director			commemorates
award received	after a work by	set in period		educated at		
		theme music		catalog		

Fig. 8. Strict relationship set

To construct the Item KG, we mapped the movies in MovieLens to the entities in the Domain KG using KB4Rec [19], a public domain linked knowledge base dataset. The KB4Rec dataset contains the mapping between the movie ID in MovieLens to the entity ID in freebase [19]. Since the Freebase-Wikidata mapping data[4] is available, MovieLens and Wikidata were integrated by connecting the two datasets. However, not all movies in MovieLens can be mapped to the Wikidata entities due to the different releases of the two datasets. Therefore, we used *movie title* and *release time* in the MovieLens dataset as keywords to complete the remaining mapping. After mapping was completed, the triples that did not contain the movies in MovieLens were removed. Since paths with a length of three or more are not helpful for generating persuasive explanations in the movie domain, we focused on paths with $d = 2$ in this experiment.

Finally, the triples of the Item KG were stored in the Neo4j graph database for further path extraction and translation processing. The paths were extracted with Python through the Neo4j APIs. Table 3 shows part of the templates used in the translation step.

[4] https://developers.google.com/freebase.

Table 3. Example of template

Relation	Template
Award received	How about __? It also received __, like the movie __ which you viewed before
Based on	__ and __ are both based on __
Inspired by	__ is inspired by __
Main subject	Remember __? __ has the same topic: __

4.3 Recommendation Modules

In order to evaluate the model-agnostic properties of the proposed system, we used two conventional recommender systems and two state-of-art recommender systems mentioned in Sect. 2.1 as the recommendation modules in our system. It is very difficult to generate high-quality personalized explanations with only these selected algorithms.

- Item k-nearest neighbor (ItemKNN) [15]
- Simple Algorithm for Recommendation (SAR) [2]
- Neural Collaborative Filtering (NCF) [9]
- Wide and Deep Model (W&D) [7]

Although we have not tested our system in combination with the existing accurate recommender systems, it is reasonable to assume that our system can be applied to any recommendation algorithms.

5 Evaluation

5.1 Mean Explainability Precision

The explainability of the proposed system is evaluated by mean explainability precision (MEP) [16]. MEP calculates the average proportion of explainable items in the top-n recommended items to the total number of recommended (top-n) items for each user to measure the precision of explainability.

$$MEP = \frac{1}{U} \sum_{u=1}^{U} \frac{N_{exp}}{L}, \tag{6}$$

where N_{exp} is the number of explainable items in the top-n recommendations, L is the recommended (top-n) items for each user, and U is the number of users.

In our experiment, an explainable item means at least one candidate path exists.

The results of the combined recommender systems and the KGs are shown in Table 4. W&D experiments could not be conducted on Slack Item KG 20m and Strict Item KG 20m due to an out-of-memory error.

As Item KG becomes sparse, MEPs decrease. However, even with Strict Item KG 20m the sparsest Item KG, the proposed system is still able to explain most of the recommended items.

Table 4. MEP@20

	ItemKNN	SAR	NCF	W&D
Slack Item KG 100k	0.8725	0.9941	0.9935	0.8928
Slack Item KG 1m	0.9167	0.9753	0.9983	0.9358
Slack Item KG 20m	0.6866	0.9516	0.9823	–
Strict Item KG 100k	0.8145	0.9595	0.9433	0.8621
Slack Item KG 1m	0.7878	0.9614	0.9760	0.8958
Strict Item KG 20m	0.5970	0.9370	0.9421	–

5.2 Case Study

The results of the case studies verified that the proposed system is able to generate more diverse and high-quality personalized explanations than the conventional model-agnostic method. The following is the result for user no. 53 in MovieLens-100k with strict relations. Table 5 shows the explanations generated by different recommendation modules. Table 6 compares the proposed approach with the existing approach, where SAR is used as the recommendation module. In both tables, the orange and blue titles represent a movie rated by a user and a recommended item, respectively.

Table 5. Explanation generated by different recommendation module

	Explanation(k=1, d=2)
ItemKNN	Living in Oblivion's cast member Dermot Mulroney is also a cast member in Capycat.
SAR	Remember The Fifth Element? Independence Day was also nominated for Satellite Award for Best Visual Effects.
NCF	The Birdcage's cast member Tim Kelleher is also a cast member in Independence Day.
W&D	Michael Kahn is the film editor of both Schindler's List and Twister.

The proposed system is clearly able to generate a higher-quality explanation containing a large volume of information. Compared to the explanations of the recommended items generated by the conventional method, the explanations generated by the proposed system are more persuasive and natural. In addition, the proposed system can create diverse and high-quality explanations even when recommending the same item to the same user.

Table 6. Explanation generated by different model-agnostic method

	Explanation(k=1, d=2)
Existing Method	User No.132 and five other users who are similar to you also watched Independence Day. Because you watched Men in Black, we recommend Independence Day.
Proposed Method	Remember The Fifth Element? Independence Day was also nominated for Satellite Award for Best Visual Effects. Remember Men in Black's cast member Will Smith? Will Smith is also a cast member in Independence Day. Independence Day and Men in Black have the same topic: extraterrestrial life.

6 Conclusion

We proposed a model-agnostic recommendation explanation model that receives a recommendation as input and generates an explanation based on the paths in a knowledge graph. The proposed system showed promise for integrating third-party knowledge bases to enhance the explainability of existing recommender systems. The results of the various evaluations indicated that the proposed system can generate high-quality personalized explanations without affecting the performance of the existing recommender system. Further analysis revealed that the proposed system can provide more diverse explanations compared with the existing model-agnostic method.

A number of factors will be examined in future research. In this paper, the rating scores were treated as equal. However, one star and two stars can be assumed to mean that the user dislikes the movie when the maximum rating is 5 stars. Therefore, the weights of the rating relation in the Item KG should be considered in a future study. Moreover, the proposed system uses templates to translate the KG paths into a natural language. Developments in natural language processing and deep learning may enable translation by text generation. Lastly, although knowledge bases are used in this paper, the reasoning is not leveraged. In the future, a reasoning process can be introduced to enhance the quality of the explanations.

Acknowledgments. This work was partly supported by JSPS KAKENHI Grant Numbers 18H03242, 18H03342, and 19H01138.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng. **17**(6), 734–749 (2005). https://doi.org/10.1109/TKDE.2005.99
2. Aggarwal, C.C.: Recommender Systems. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-29659-3
3. Ai, Q., Azizi, V., Chen, X., Zhang, Y.: Learning heterogeneous knowledge base embeddings for explainable recommendation. ArXiv arXiv:1805.03352 (2018)
4. Balog, K., Radlinski, F., Arakelyan, S.: Transparent, scrutable and explainable user models for personalized recommendation. In: SIGIR 2019, pp. 265–274 (2019)
5. Boschin, A., Bonald, T.: WikiDataSets: standardized sub-graphs from WikiData. arXiv:1906.04536 [cs, stat], June 2019. http://arxiv.org/abs/1906.04536
6. Catherine, R., Mazaitis, K., Eskénazi, M., Cohen, W.W.: Explainable entity-based recommendations with knowledge graphs. ArXiv arXiv:1707.05254 (2017)
7. Cheng, H.T., et al.: Wide & deep learning for recommender systems. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems DLRS 2016, pp. 7–10. Association for Computing Machinery, New York (2016)
8. Gong, Y., Zhang, Q.: Hashtag recommendation using attention-based convolutional neural network. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence IJCAI 2016, pp. 2782–2788. AAAI Press (2016)
9. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web WWW 2017, pp. 173–182. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE (2017)
10. Ma, W., et al.: Jointly learning explainable rules for recommendation with knowledge graph. In: WWW 2019, pp. 1210–1221 (2019)
11. Peake, G., Wang, J.: Explanation mining: post hoc interpretability of latent factor models for recommendation systems. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining KDD 2018, pp. 2060–2069. Association for Computing Machinery, New York (2018)
12. Singh, J., Anand, A.: Posthoc interpretability of learning to rank models using secondary training data. ArXiv arXiv:1806.11330 (2018)
13. Tintarev, N., Masthoff, J.: Designing and evaluating explanations for recommender systems. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 479–510. Springer, Boston, MA (2011). https://doi.org/10.1007/978-0-387-85820-3_15
14. Wang, H., et al.: RippleNet: propagating user preferences on the knowledge graph for recommender systems. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management CIKM 2018, pp. 417–426. Association for Computing Machinery, New York (2018)
15. Wang, J., de Vries, A.P., Reinders, M.J.T.: Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval SIGIR 2006, pp. 501–508. Association for Computing Machinery, New York, NY (2006)
16. Wang, S., Tian, H., Zhu, X., Wu, Z.: Explainable matrix factorization with constraints on neighborhood in the latent space. In: Tan, Y., Shi, Y., Tang, Q. (eds.) DMBD 2018. LNCS, vol. 10943, pp. 102–113. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93803-5_10

17. Wang, X., Chen, Y., Yang, J., Wu, L., Wu, Z., Xie, X.: A reinforcement learning framework for explainable recommendation. In: 2018 IEEE International Conference on Data Mining (ICDM), pp. 587–596, November 2018. https://doi.org/10.1109/ICDM.2018.00074
18. Zhang, Y., Chen, X.: Explainable recommendation: a survey and new perspectives. ArXiv arXiv:1804.11192 (2018)
19. Zhao, W.X., et al.: Kb4rec: a data set for linking knowledge bases with recommender systems. Data Intell. **1**(2), 121–136 (2019)