Explainable Recommendation Using Knowledge Graphs and Random Walks

Kaname Muto
Hokkaido University
kaname-0-sugar@eis.hokudai.ac.jp

Satoshi Oyama
Hokkaido University
oyama@ist.hokudai.ac.jp

Itsuki Noda
Hokkaido University
i.noda@ist.hokudai.ac.jp

***Abstract*—A knowledge graph (KG) contains rich information about users and items. The relationship among users and items can help to generate intuitive explanations for recommended items. Many variations of KG-based recommendation algorithms use the shortest path from the user to the item in order to generate an explanation of the recommendation. However, the simple shortest path may not be useful in the case when the path is long, because the interpretation of the long path is difficult. Also, there may be no path between the user and the recommended item. In order to overcome these difficulties, we proposed an extension of the existing framework based on random walk with KG embedding. In the proposed framework, we use the most probable path in a random walk as an explanation. Thereby, our framework can even explain items that have no connection in the KG due to the latent connection resulting from random walk teleportation. Comparison experiment demonstrated that the framework can provide more suitable recommendations than the existing method. In addition, the experiment show the ability of the proposed method to generate explanation for all recommendations that have no path in the graph.**

***Index Terms*—Recommendation, Knowledge Graph Embedding, Random Walk, Explainability**

I. INTRODUCTION

With the large amount of data that has accumulated over time, it can be difficult for users to find content of interest to them. To improve the user experience, almost every service that provides content to users is equipped with a recommendation system.

Among the many approaches to recommendation systems, knowledge graphs (KGs) are one of the most effective. The users, items, and their attribute information in the recommendation problem can be represented as a KG by connecting nodes with appropriate relationships. Prior studies have shown that a KG-based recommendation system can comprehensively handle users, items, and their attribute information, reflect the various relationships between users and items, and improve the performance of the recommendation algorithm [1] [2]. In addition, the paths across multiple nodes from user to item have explainability and interpretability [3] [4].

Explainability and interpretability have become increasingly important in recommendation systems [5]. An explainable recommendation system provides recommendation results in addition to explanations to clarify why the items were recommended. This is expected to improve system transparency, effectiveness, user satisfaction, and reliability. It should also contribute to the system designer's ability to diagnose, debug, and improve the recommendation algorithm.

In general, for models with high recommendation accuracy, the reason for the recommendations are often not intuitive and difficult to understand. Xie et al.'s study [6] used a multi-objective optimization method that optimizes conflicting objectives simultaneously. This method uses a KG, and the path from the user to the recommended item is used as the explanation. In their study, explainability was defined quantitatively as an objective function. However, only items that were actually connected in the KG were candidates for recommended items, and items with no connection could not be recommended.

Wang et al. [7] developed a method that uses an attention mechanism for the KG. By showing the paths that follow important relationships rather than the shortest path, it is possible to show the reason for the recommendation, such as whether the brand of the item or having purchased a similar item is important. However, depending on the item to be recommended, the path from the user to the recommended item may be long, resulting in low interpretability, or the absence of a path from the user to the recommended item may make it impossible to provide an explanation.

The random walk method is effective for recommending and explaining items in the KG that are at a large distance from the user or not connected to the user. The stationary distribution of a random walk of the KG is the importance of each node, which is used for recommendation. A random walk can associate items that are not directly connected by conveying information along the edges of the KG. It thereby provides a wide coverage of candidate recommendations. In addition, latent connection by teleportation allows the path from user to item to be used as an explanation.

Nikolakopoulos et al. proposed a framework, RecWalk [8], that combines random walk with an arbitrary algorithm. In Nikolakopoulos et al.'s study, they use a KG composed of only users and items. RecWalk uses the Sparse Linear Method (SLIM) [9] to obtain item similarity and recommends the top n items by random walk based on the similarity. It has been reported to be more accurate than models based on deep learning.

Suzuki et al. [10] extended this framework and proposed a method that combines random walk and KG embedding by using a KG that includes not only users and items but also attribute information of items. Suzuki et al.'s framework uses KG embedding to obtain item similarity and recommends the top n items by random walk based on the similarity. The accuracy was slightly higher than that of RecWalk and other existing methods.

In this study, we introduce the function of explaining recommendations into Suzuki et al.'s framework. We use KG embedding to compute item similarity and random walk to recommend the top n items. For explanation, we provide the shortest paths in a KG and the most probable path during a random walk. On the Luxury Beauty dataset, our framework demonstrated higher recommendation accuracy than that of existing methods, and we showed that the explainability and interpretability were improved through comparisons of shortest paths and most probable paths.

II. PRELIMINARIES

A. Knowledge Graph

A knowledge graph is a directed graph in which nodes represent entities $e \in \mathcal{E}$, while edges in the graph function as relations $r \in \mathcal{R}$ between entities. Each edge is represented in the triplet (head entity, relation, tail entity), implying the specific relationship between the head entity and tail entity. We define a knowledge graph as $\mathcal{KG} = \{(h, r, t) \mid h, t \in \mathcal{E}, r \in \mathcal{R}\}$. In this study, users, items, and their attribute information are represented as entities. For example, when a user u clicks or purchases an item i, the triplet is represented as $(u, interact, i)$.

B. Knowledge Graphs Embedding

Knowledge graph embedding is a mapping of entities on $\mathcal{KG}$ to a low-dimensional vector space. The embedded vectors can retain their graph structure from their distance and inner product in the vector space. In this study, we used TransE [11] as the embedding method. Given a set D of triplets (h, r, t), we want $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ to be satisfied for all triplets as much as possible. Conversely, we want to make sure that $\mathbf{h}+\mathbf{r} \approx \mathbf{t}$ does not hold for any triplet not in KG. In TransE embedding, the KG entities are embedded to minimize the following objective function.

$$L = \sum_{(h,r,t)\in D} \sum_{(h',r,t')\in D'} \gamma + d(\mathbf{h} + \mathbf{r}, \mathbf{t}) - d(\mathbf{h}' + \mathbf{r}, \mathbf{t}'), \tag{1}$$

where $\gamma > 0$ is a margin hyperparameter, $d(.,.)$ is Euclidean distance as the distance function, and

$$D' = \{(h', r, t) \mid h' \in \mathcal{E}\} \cup \{(h, r, t') \mid t' \in \mathcal{E}\}$$

is the negative sample.

C. Random Walk

In this study, Personalized PageRank (PPR) is used to introduce into the recommendation algorithm the idea in web information retrieval that a web page is important if other important pages point to it. When considering a recommendation to user u, a random walker starts from a node of user u in the KG and randomly transitions to the adjacent nodes with probability μ. We also assume that the random walker can teleport to the node of user u with probability $1 - \mu$. The stationary distribution of this random walk process represents the importance of each node, and the item corresponding to the top n values of the stationary vector is the item to recommend. Let $\boldsymbol{P} \in \mathbb{R}^{N\times N}$ be the transition probability matrix of the KG. The stationary distribution $\boldsymbol{p}_u \in \mathbb{R}^N$ is obtained as follows:

$$\boldsymbol{p}_u = \lim_{K\to\infty} \boldsymbol{e}_u^\top (\mu \boldsymbol{P} + (1-\mu)\mathbf{1}\boldsymbol{e}_u^\top)^K. \tag{2}$$

The vector $\boldsymbol{e}_u^\top \in \mathbb{R}^N$ is a personal vector of user u for teleporting to itself. The transition probability matrix $\boldsymbol{P}$ is usually

$$\boldsymbol{P} = diag(\boldsymbol{A}\mathbf{1})^{-1}\boldsymbol{A}, \tag{3}$$

where $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ is the adjacency matrix of the KG.

III. FRAMEWORK

This section describes the recommendation framework combining the random walk method with KG embedding methods. The framework is an extension of RecWalk [8] and was proposed in a previous study [10]. The similarity of entities is computed by KG embedding, and the top n recommendations are made by random walk. For each recommended item, the shortest path in the KG and the most probable path by random walk are used for explanation.

A. Random Walk with KG Embedding

Consider a random walk model that transitions to adjacent nodes and teleports with certain probabilities. In this case, the choice of the node to teleport to is usually random, but the feature of this method is that it takes into account the similarity of entities. The similarity of the nodes is the inner product of the embedding vectors obtained by KG embedding. For the vector $\boldsymbol{v}_i (i = 1, ..., N)$ obtained by embedding, we define a matrix $\boldsymbol{M}$ representing the similarity of the nodes as

$$M_{ij} = \boldsymbol{v}_i^\top \boldsymbol{v}_j. \tag{4}$$

To make this matrix a transition probability matrix, we use the stochasticity adjustment strategy according to [10] to define $\boldsymbol{M}_n$. In this way, using the matrix $\boldsymbol{M}_n$ representing the similarity of the entities obtained by KG embedding and the adjacency matrix $\boldsymbol{A}$ of the KG, the transition probability matrix is calculated as follows.

$$\boldsymbol{P} = \alpha \cdot diag(\boldsymbol{A}\mathbf{1})^{-1}\boldsymbol{A} + (1-\alpha)\boldsymbol{M}_n, \tag{5}$$

where the parameter α represents the probability of transitioning on the basis of the adjacent node, and teleportation is performed considering the similarity of entities with probability $1 - \alpha$. Using this transition probability matrix, the top n items are recommended by the PPR in Eq. (2) for each user.

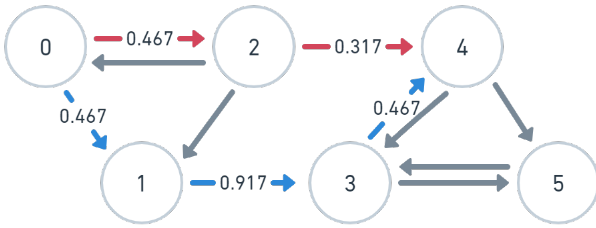

Fig. 1. Example of paths from node 0 to node 4

TABLE I
NUMBER OF KG ENTITIES AND RELATIONS

		LB4	AF3
entity	user	5422	3100
	item	2119	1062
	brand	3	332
relation	user *interact* item	36688	12629
	item $interact^{-1}$ user	36688	12629
	item *belong to* brand	3	696
	item *also view* item	696	3
	item *also buy* item	829	15

B. Explanation

For the explanations of the top n recommended items for user u, we use the shortest paths in the adjacency relation of the knowledge graph and the most probable paths in a random walk.

The paths have a weight of 1 if the nodes are directly connected, and a very large value if they are not. The shortest path is the one with the smallest sum of weights among the paths starting at user u and ending at each recommended item. Since edges have no negative weights, this problem can be solved by Dijkstra's method.

The most probable path in a random walk is the path with the highest probability in the transition from user u to each recommended item. That is, the path that maximizes the product of the probabilities of each transition from user u to the recommended item. To solve this problem, we first take the logarithm of each element of the transition probability matrix obtained by Eq. (5). Then, each element is multiplied by -1 to obtain a positive value, which can be considered as a weight. In other words, each element P_{ij} of the transition probability matrix $\boldsymbol{P}$ is replaced by the weight matrix $\boldsymbol{W}$ as follows:

$$W_{ij} = -\log P_{ij}. \tag{6}$$

This means that instead of maximizing the product of probabilities, the sum of weights is minimized, and since there are no negative weights, this problem can also be solved by Dijkstra's method.

Fig. 1 shows an example of the shortest path and the most probable path. From the adjacency matrix $\boldsymbol{A}$ of this KG, the transition probability matrix is computed as in Eq. (5). In this example, $\alpha = 0.9$, and the teleportation matrix $\boldsymbol{M}_n$ is assumed to be uniformly random. The paths from node 0 to node 4 are shown with a red line for the shortest path and a blue line for the most probable path. As this example shows, the shortest path and the probable path are sometimes different, and each can be interpreted and explained differently.

The obtained paths are displayed as an explanation. It is also possible to interpret the reason for the recommendation. For example, if the route "user, item1, item2" recommends item2 to the user, it can be interpreted as "item2 is similar to item1 that you bought".

IV. EXPERIMENTS

A. Dataset

In this study, we used Luxury Beauty 4-core (LB4) and AMAZON FASHION 3-core (AF3) from Amazon Review Data [12]. These data have been reduced to extract the k-core such that each of the remaining users and items have k reviews each. Using these datasets, we created KGs for each. The number of entities and relations obtained are summarized in TABLE I. A quarter of each data set was used as test data, and cross-validation and hyperparameter tuning were performed on the remaining data.

B. Evaluation

The following methods were used to compare the performance. We evaluated the performance of the top 10 recommended items using Mean Average Precision (MAP) [13].

For top n recommended items, MAP is to average the Average Precision (AP) over all $|U|$ users.

$$MAP@n = \frac{1}{|U|} \sum_{u \in U} AP_u@n, \tag{7}$$

$$AP@n = \frac{1}{m} \sum_{k=1}^{n} Precision@k \cdot rel(k), \tag{8}$$

where m is the number of relevant items and $rel(k)$ is an indicator that k^{th} item was relevant ($rel(k) = 1$) or not ($rel(k) = 0$).

- **PPR**
  Random walk model as a baseline
- **TransE** [11]
  KG embedding model
- **NFM** [14]
  Neural Factorization Model (NFM) as a baseline for the DL-based models
- **RecWalk** [8]
  Random walk with SLIM [15]
- **Proposed** [10]
  Random walk with KG embedding. We extended the function of explaining recommendations.

TABLE II shows the results of the methods compared. On the Luxury Beauty dataset, the proposed method outperformed the other methods. However, RecWalk outperformed the other methods on the AMAZON FASHION dataset. One possible

TABLE II
MAP SCORES

	PPR	TransE	NFM	RecWalk	Proposed
LB4	0.05350	0.04224	0.02245	0.06012	**0.12121**
AF3	0.11150	0.11960	0.08243	**0.14720**	0.13403

reason for this could be the influence of the number of relationships in the KG. The LB4 dataset contains more item-to-item relationships and fewer item-to-brand, while there are more item-to-brand relationships and fewer item-to-item in the AF3 dataset. Item-item relationships are considered more suitable than item-brand relationships for obtaining similarity through KG embedding. However, in terms of explainability, relationships about the attribute information of these items can be used for explanation, which is an advantage over RecWalk.

C. Explainability

Our method presents the shortest paths and most probable paths for the top 10 recommended items. TABLE III and TABLE IV summarize the paths for each dataset. Concerning the length of the paths, the most probable path is longer, but both are less than 3. In Xie et al.'s study [6], user satisfaction decreases as the path to the recommended item becomes longer, so the length of the path should be less than 3 in order for the explanation to be effective.

Next, we consider the case where each path is different. The percentages of different paths are 5.01% and 12.43% for LB4 and AF3, respectively. Even if there is no path from the user to the recommended item in the KG, the most probable path can provide some explanation by teleportation. There are cases where the shortest past is different from the most probable path. Two examples are shown in Fig. 2 and Fig. 3. The red line is shortest path and the blue dotted line is the most probable path. In Fig. 2, the user is recommended "Foundation", and the shortest path goes through another "Foundation", while the most probable path goes through "Sun Damage Repair" and another user. Similarly, in Fig. 3, "Shaving Cream" is recommended to the user, and the shortest path goes through another "Shaving Cream", while the most probable path goes through "Razor" and another user. The shortest path can be interpreted as "an item similar to the item you bought", while the most probable path can be interpreted as "people who bought the item you bought also bought this". In other words, it can be explained as either a similar item or as an item that is likely to be bought at the same time.

In addition, the shortest path is likely to go through popular items with many edges. However, the most probable path is less likely to go through popular items because the path through popular items, which have many edges, has low-probability in the random walk. Therefore, it is possible for an explanation to not rely on popular items.

TABLE III
LUXURY BEAUTY 4-CORE

	Average length	NoPath [%]
Shortest path	2.376	3.24
Probable path	2.515	0.00

TABLE IV
AMAZON FASHION 3-CORE

	Average length	NoPath [%]
Shortest path	2.597	9.87
Probable path	2.688	0.00

Fig. 2. Paths from user 1585 to item 676

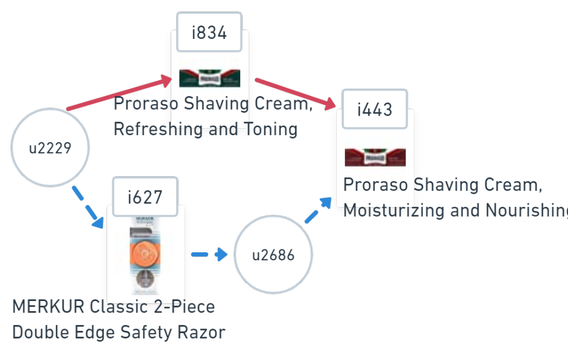

Fig. 3. Paths from user 2229 to item 443

V. CONCLUSION

In this study, we extended the recommendation framework combining the random walk method with KG embedding for explainable recommendations. This framework outperformed other methods on the Luxury Beauty dataset and was comparable to RecWalk on the AMAZON FASHION dataset. The shortest path and the most probable path could be displayed for explanation, each of which could be interpreted differently. The use of the most probable path also demonstrated that all items can be explained, even if no paths exist in the KG.

VI. Acknowledgments

This work was partially supported by JSPS KAKENHI Grant Number JP18H03337 and JST CREST Grant Number JPMJCR21D1.

References

[1] X. Yu, X. Ren, Y. Sun, Q. Gu, B. Sturt, U. Khandelwal, B. Norick, and J. Han, "Personalized entity recommendation: A heterogeneous information network approach," in *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, 2014.

[2] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, "Collaborative knowledge base embedding for recommender systems," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016.

[3] Q. Ai, V. Azizi, X. Chen, and Y. Zhang, "Learning heterogeneous knowledge base embeddings for explainable recommendation," *Algorithms*, 2018.

[4] F. Yang, N. Liu, S. Wang, and X. Hu, "Towards interpretation of recommender systems with sorted explanation paths," in *2018 IEEE International Conference on Data Mining*, 2018.

[5] Y. Zhang, X. Chen *et al.*, "Explainable recommendation: A survey and new perspectives," *Foundations and Trends® in Information Retrieval*, vol. 14, no. 1, pp. 1–101, 2020.

[6] L. Xie, Z. Hu, X. Cai, W. Zhang, and J. Chen, "Explainable recommendation based on knowledge graph and multi-objective optimization," *Complex & Intelligent Systems*, vol. 7, no. 3, pp. 1241–1252, 2021.

[7] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "Kgat: Knowledge graph attention network for recommendation," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery and data mining*, 2019, pp. 950–958.

[8] A. N. Nikolakopoulos and G. Karypis, "Recwalk: Nearly uncoupled random walks for top-n recommendation," *Proceedings of the 12th ACM International Conference on Web Search and Data Mining*, 2019.

[9] X. Ning and G. Karypis, "Slim: Sparse linear methods for top-n recommender systems," in *2011 IEEE 11th International Conference on Data Mining*, 2011, pp. 497–506.

[10] T. Suzuki, S. Oyama, and M. Kurihara, "A framework for recommendation algorithms using knowledge graph and random walk methods," in *Proceedings of the 4th IEEE Workshop on Human-in-the-Loop Methods and Future of Work in BigData*, 2020.

[11] A. Bordes, N. Usunier, A. Garcia-Durán, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, 2013.

[12] J. Ni, J. Li, and J. McAuley, "Justifying recommendations using distantly-labeled reviews and fine-grained aspects," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 2019, pp. 188–197.

[13] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval: The Concepts and Technology behind Search*, 2nd ed. USA: Addison-Wesley Publishing Company, 2011.

[14] X. He and T.-S. Chua, "Neural factorization machines for sparse predictive analytics," *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017.

[15] X. Ning and G. Karypis, "Slim: Sparse linear methods for top-n recommender systems," in *2011 IEEE 11th International Conference on Data Mining*, 2011, pp. 497–506.