KB4Rec: A Dataset for Linking Knowledge Bases with Recommender Systems

Wayne Xin Zhao, Gaole He, Hongjian Dou, Jin Huang, Siqi Ouyang and Ji-Rong Wen
School of Information, Renmin University of China
{batmanfly,ouyangsiqi0726}@gmail.com,{hegaole,hongjiandou,jin.huang,jrwen}@ruc.edu.cn

ABSTRACT

To develop a knowledge-aware recommender system, a key data problem is how we can obtain rich and structured knowledge information for recommender system (RS) items. Existing datasets or methods either use side information from original recommender systems (containing very few kinds of useful information) or utilize private knowledge base (KB). In this paper, we present the first public linked KB dataset for recommender systems, named *KB4Rec v1.0*, which has linked three widely used RS datasets with the popular KB Freebase. Based on our linked dataset, we first preform some interesting qualitative analysis experiments, in which we discuss the effect of two important factors (*i.e.,* popularity and recency) on whether a RS item can be linked to a KB entity. Finally, we present the comparison of several knowledge-aware recommendation algorithms on our linked dataset.

KEYWORDS

Knowledge-aware recommendation, recommender systems, knowledge base

ACM Reference format:
Wayne Xin Zhao, Gaole He, Hongjian Dou, Jin Huang, Siqi Ouyang and Ji-Rong Wen. 2018. KB4Rec: A Dataset for Linking Knowledge Bases with Recommender Systems. In *Proceedings of ACM conference, Vancouver, Canada., Opctober 2018 (RecSys'18),* 5 pages.
DOI: 10.475/123_4

1 INTRODUCTION

Nowadays, recommender systems (RS), which aim to match users with interested items, have played an important role in various online applications. Traditional recommendation algorithms mainly focus on learning effective preference models from historical user-item interaction data, *e.g.,* matrix factorization [12]. With the rapid development of Web techniques, various kinds of side information has become available in RSs, called *context* [14]. In an early stage, such context information is usually unstructured, and its availability is limited to specific data domains or platforms.

Recently, more and more efforts have been made by both research and industry communities for structurizing world knowledge or domain facts in a variety of data domains. One of the most typical organization forms is *knowledge base (KB)* [22], also called knowledge graph. KBs provide a general and unified way to organize and relate information entities, which have been shown to be useful in many applications. Specially, KBs have also been used in recommender systems, called *knowledge-aware recommender systems* [2]. To develop a knowledge-aware recommender system, a key data problem is how we can obtain rich and structured knowledge information for RS items. Overall, there are two main solutions from existing studies. First, they collect side information from the RS platform [6, 8, 16], and some studies may further construct tiny and simple KB-like knowledge structure [4, 23]. The number of attributes or relations is usually small, and much useful world knowledge is likely to be missed. Second, several works propose to link RS with private KBs [20, 21, 24]. The linkage results are not publicly available.

To address the need for the linked dataset of RS and KBs, we present the first public linked KB dataset for recommender systems, named *KB4Rec* v1.0, freely available at https://github.com/RUCDM/KB4Rec. Our basic idea is to heuristically link items from RSs with entities from a public large-scale KB[1]. On the RS side, we select three widely used datasets (*i.e.,* MovieLens [6], LFM-1b [16] and Amazon book [8]) covering three different data domains, namely movie, music and book; on the KB side, we select the well-known Freebase [5]. We try to maximize the applicability of our linked dataset by selecting very popular RS datasets and KBs. We do not share the original datasets, since they are maintained by original researchers or publishers.

In our KB4Rec v1.0 dataset, we organized the linkage results by linked ID pairs, which consists of a RS item ID and a KB entity ID. All the IDs are inner values from the original datasets. Once such a linkage has been accomplished, it is able to reuse existing large-scale KB data for RSs. For example, the movie of "Avatar" from MovieLens dataset [6] has a corresponding entity entry in Freebase, and we are able to obtain its attribute information by reading out all its associated relation triples in Freebase. Based on the linked dataset, we first preform some interesting qualitative analysis experiments, in which we discuss the effect of two important factors (*i.e.,* popularity and recency) on whether a RS item can be linked to a KB entity. Finally, we present the comparison of several knowledge-aware recommendation algorithms on our linked dataset.

To our knowledge, it is the first public dataset for linking KBs with RSs. With our linkage results and original data copies, it is easy to develop an evaluation set for knowledge-aware recommendation algorithms. We believe such a dataset is beneficial to the development of knowledge-aware recommender systems.

2 EXISTING DATASETS AND METHODS

In this section, we briefly review the related datasets and methods.

[1]We use the terms of "items" and "entities" respectively for RSs and KBs.

RecSys'18, Vancouver, Canada.

DOI: 10.475/123_4

Early knowledge-aware recommendation algorithms are also called context-aware recommendation algorithms, in which the side information from the original RS platform is considered as context data. For example, social network information of Epinions dataset is utilized in [11, 13], POI property information of Yelp dataset is utilized in [4], movie attribute information of MovieLens dataset is utilized in [23] and user profile information of microblogging dataset has been utilized in [25]. These datasets usually contain very few kinds of side information, and the relation between different kinds of side information is ignored.

To make such side information more structured, Heterogeneous Information Networks (HIN) have been proposed as a general technique for modeling information networks [19]. In HINs, we can effectively learn underlying relation patterns (called *meta-path*) and organize side information via meta-path-based representations. For example, HIN-based recommendation have been applied to solve PER [23], HeteRecom [18] and MCRec [9]. HIN based algorithms usually rely on graph search algorithms, which is difficult to deal with large-scale relation pattern finding.

More recently, KBs have become a popular kind of data resources to store and organize world knowledge or domain facts. Many studies have been proposed [22] for the construction, inference and applications of KBs. Specially, several pioneering studies try to leverage existing KB information for improving the recommendation performance [20, 21, 24]. They apply a heuristic method for linking RS items with KB entities. In these studies, they use a private KB for linkage, which cannot be obtained publicly.

3 LINKED DATASET CONSTRUCTION

In our work, we need to prepare two kinds of datasets, namely RS and KB. Next, we first give the detailed descriptions of the original datasets from two aspects, and then discuss the linkage method.

RS Datasets. We consider three popular RS datasets for linkage, namely MovieLens, LFM-1b and Amazon book, which covers the three domains of movie, music and book respectively.

- *MovieLens* dataset [6] describes users' preferences on movies. A preference record takes the form ⟨user, item, rating, timestamp⟩, indicating the rating score of a user on a movie on some time. There have been four MovieLens datasets released, known as 100K, 1M, 10M, and 20M, reflecting the approximate number of ratings in each dataset. We select the largest MovieLens 20M for linkage.
- *LFM-1b* dataset [16] describes users' interaction records on music. It provides information including artists, albums, tracks, and users, as well as individual listening events. It records the listening count of a song by a user, but does not contain rating information.
- *Amazon book* dataset [8] describes users' preferences on book products, which has a data form, *i.e.,* ⟨user, item, rating, timestamp⟩. The dataset is very sparse, containing 22 million ratings from 8 million users across nearly 23 million items.

The three datasets all provide several kinds of side information such as item titles (all), IMDB ID (movie), writer (book) and artist (music). We utilize such side information for subsequent KB linkage.

KB Dataset. We adopt the large-scale pubic KB *Freebase*. Freebase [5] is a KG announced by Metaweb Technologies, Inc. in 2007 and was acquired by Google Inc. on July 16, 2010. Freebase stores facts by triples of the form ⟨head, relation, tail⟩. Since Freebase shut down its services on August 31, 2016, we use the version of March 2015, which is its latest public version.

RS to KB Linkage. All three RS datasets provide the information of item titles. With an offline Freebase search API, we retrieve KB entities with item titles as queries. Our heuristic linkage method follows the similar idea in [17]. If no KB entity with the exactly same title was returned, we say the RS item is *rejected* in the linkage process. If at least one KB entity with the exactly same title was returned, we further incorporate one kind of side information as a refined constraint for accurate linkage: *IMBD ID*, *artist name* and *writer name* are used for the three domains of movie, music and book respectively. We have found only a small number (about one thousand for each domain) of RS items can not be accurately linked or rejected via the above procedure, and we simply discard them. During the linkage process, we have dealt with several problems that will affect the results of string match algorithms, *e.g.,* lowercase, abbreviation, and the order of family/given names. Since the LFM-1b dataset is extremely large, we remove all the musics with fewer than ten listening events. Even after filtering, it still contains about 6.5 million musics.

Basic Statistics. We summarize the basic statistics of the three linked datasets in the second column of Table 1. It can be observed that for the MovieLens 20M dataset, we have a very high linkage ratio: about 95.2% items can be accurately linked to a KB entity. For LFM-1b dataset, the linkage ratio is 19.4%. But, the linkage ratio for the book domain is very low, about 4.7%. A possible reason that MovieLens 20M dataset has a very high linkage ratio is that it contains fewer items than the other two datasets, which themselves are refined by original releasers. Besides, we speculate that there may be some domain bias in the construction of Freebase. Although the linkage ratios for the latter two datasets are not high, the absolute numbers of linked items are large. Such a linked dataset is feasible for research-purpose studies.

Shared Datasets. We name the above linked linked KB dataset for recommender systems as *KB4Rec* v1.0, freely available at https://github.com/RUCDM/KB4Rec. In our KB4Rec v1.0 dataset, we organized the linkage results by linked ID pairs, which consists of a RS item ID and a KB entity ID. All the IDs are inner values from the original datasets. We have 25,982, 1,254,923, and 109,671 linked ID pairs for MovieLens 20M, LFM-1b and Amazon book respectively.

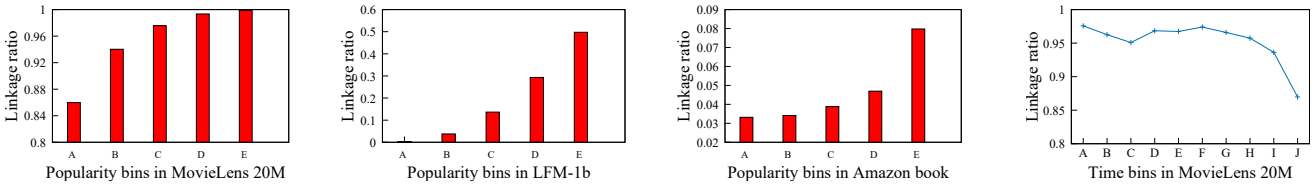
4 LINKAGE ANALYSIS

Previously, we have shown the linkage ratios for different datasets. We find that a considerable amount of RS items can not be linked to KB entities. It is interesting to study what factors will affect the linkage ratio. We consider two kinds of factors for analysis.

Effect of Popularity on Linkage. Intuitively, a popular RS item should be more likely to be included in a KB than an unpopular item, since it is reasonable to incorporate more "important" RS items judged by the RS users into KBs. The construction of KB

Table 1: Statistics of both original datasets and our evaluation datasets.

Datasets	Original				Evaluation		
	#Items	#Linked-Items	#Users	#Interactions	#Users	#Items	#Interactions
MovieLens 20*M*	27,279	25,982	8,026,324	20,000,264	61,583	19,533	5,868,015
LFM-1b	6,479,700	1,254,923	120,317	1,021,931,544	7,694	30,658	203,975
Amazon book	2,330,066	109,671	3,468,412	22,507,155	65,125	69,975	828,560



(a) Popularity. (b) Recency.

Figure 1: Examining the effect of two factors on the linkage results. We use *A*, *B*, $\cdots$ to indicate the bin number in an ordered way. The first three subfigures correspond to the popularity analysis, and the last one corresponds to the recency analysis.

itself usually involves manual efforts, which is difficult to avoid the bias of human attention. To measure the popularity of a RS item, we adopt a simple frequency-based method by counting the number of users who have interacted with the item. This measure characterizes the attractiveness of an item from the users in a RS. First, we sort the items ascendingly according to its popularity value. Then, we further equally divide all the users into five ordered bins with the same number of users. In this way, a user with a rank of r will be assigned to the $\lfloor \frac{r}{20} \rfloor$-th bin. Hence, an item with a larger bin number will be more popular than another with a smaller bin number. Then we compute the linkage ratio for each bin and the results are reported in Fig. 1(a) (the three subfigures on the left). It can be observed that a bin with a larger number has a higher linkage ratio than the ones with a smaller number. The results indicate that popularity is likely to have positive effect on linkage.

Effect of Recency on Linkage. The second factor we consider is the recency, *i.e.,* the time when a RS item was created. Our assumption is that if a RS item was created or released on an earlier time, it would be more probable to be included in KBs. Since human attention aggregation is a gradually growing process, a RS item usually requires a considerable amount of time to become popular. To check this assumption, we need to obtain the release date of RS items. However, only the MovieLens 20*M* dataset contains such an attribute information, we only report the analysis result on this dataset. We first sort the items according to their release dates ascendingly, and then equally divide all the users into ten ordered bins following the procedure of the above popularity analysis. Finally, we compute the linkage ratios for each bin. The results are reported in Fig. 1(b). We can see that the linkage ratios gradually decrease with time going. The results indicate that recency is likely to have negative effect on linkage, *i.e.,* an older RS item seems to be more probable to be included in a KB than a more recent one. The last bin has a dramatic drop, since our version of Freebase is March 2015.

5 EXPERIMENT

In this section, we present the comparison of some existing recommendation algorithms on our linked datasets.

Experimental Setup. Since our linked datasets are very large, we first generate a small test set for the following experiments. We take the subset from the last year for LFM-1b dataset and the subset from year 2005 to 2015 for MovieLens 20*M* dataset. We also perform 3-core filtering for Amazon book dataset and 10-core filtering for other datasets. The statistics of dataset used in [10] is reported in Table 1 (the last column). Following [7], we consider the last-item recommendation task for evaluation. We set up such a task since it is a commonly used evaluation setting for RSs, and it is easy to compare different methods. Given a user, first we sort the items according to the interaction timestamp ascendingly, then we take the last item into the test set and the rest into training set. The final goal is to predict the last item given the previous interaction sequence of a user. Since enumerating all the items as candidate is time-consuming, we pair each ground-truth with 100 negative items to form a randomly ordered list. Then each comparison method is to return a ranked list according its recommendation confidence. To evaluate different methods, we adopt a variety of evaluation metrics, including the Mean Reciprocal Rank (MRR), Hit Ratio (HR), and Normalized Discounted cumulative gain (NDCG).

KB Information Representation. Our focus is to provide rich knowledge information for recommender systems. A simple way is to represent KB information with a one-hot vector, which is sparse and large. Here we borrow the idea in [1, 24] to embed KB data into low-dimensional vectors. Then the learned embeddings are used for subsequent recommendation algorithms. To train TransE [1], we start with linked entities as seeds and expand the graph with one-step search. Not all the relations in KBs are useful, we remove unfrequent relations with fewer than 5,000 triples. After that, each linked item is associated with a learned KB embedding vector.

Methods to Compare. We consider the following methods for performance comparison[2]:

- **BPR** [15]: It learns a matrix factorization model by minimizing the pairwise ranking loss in a Bayesian framework.
- **SVDFeature** [3]: It is a model for feature-based collaborative filtering. In this paper we use the KB embeddings as context features to feed into SVDFeature.
- **mCKE** [24]: It first proposes to incorporate KB and other information to improve the recommendation performance. For fairness, we implement a simplified version of CKE by only using KB information, and exclude image and text information. Different from the original CKE, we fix KB representations and adopt the learned embeddings by TransE.
- **KSR** [10]: It is a Knowledge-enhanced Sequential Recommender (KSR). It incorporates KB information to enhance the semantic representation memory networks.

Results and Analysis. The results of different methods for the last-item recommendation are presented in Table 2. We can see that:

(1) Among all the methods, BPR performs worst on the first two datasets, but very well on the Amazon book dataset. A possible reason is the first two datasets are relatively dense while the Amazon book dataset is sparse. A lightweight method is likely to obtain a better performance than more complicated methods on a sparse dataset.

(2) SVDFeature is implemented with a pairwise ranking loss function, and it can be roughly understood as an enhanced BPR model with the incorporation of the learned KB embeddings. Compared with BPR, SVDFeature is slightly better on the MovieLens 20*M* dataset, substantially better on the LFM-1b dataset, but worse on the Amazon book dataset. In SVDFeature, each additional context feature will increase some number of parameters (deciding on the number of dimensions). Hence, on a sparse dataset, it may not work better than the simple BPR model.

(3) Next, we analyze the performance of the knowledge-aware recommendation methods, namely mCKE and KSR. Overall, mCKE does not work well as expected, which only gives good performance on the LFM-1b dataset. A possible reason is that our implementation of mCKE fixes the learned KB embeddings, while the original CKE model adaptively updates KB embeddings. As a comparison, the recently proposed KSR method works best consistently on the three datasets. KSR combines the capacity of modeling data sequences from Recurrent Neural Networks (RNN) and the capacity of storing data in a long term from Memory Networks (MN). It further enhances MNs with the learned KB embeddings.

Table 2: Performance comparison of different methods on the task of last-item recommendation.

Datasets	Methods	MRR	Hit@10	NDCG@10
MovieLens 20*M*	BPR	0.198	0.443	0.237
	SVDFeature	0.202	0.450	0.242
	mCKE	0.178	0.382	0.209
	KSR	0.294	0.571	0.344
LFM-1b	BPR	0.328	0.523	0.360
	SVDFeature	0.337	0.544	0.373
	mCKE	0.371	0.541	0.399
	KSR	0.427	0.607	0.460
Amazon book	BPR	0.272	0.548	0.323
	SVDFeature	0.264	0.544	0.315
	mCKE	0.248	0.494	0.291
	KSR	0.353	0.653	0.413

6 CONCLUSION

This paper introduced a public dataset for linking RS with KB, namely *KB4Rec v1.0*. Our dataset covered three domains consists of a large number of linked ID pairs. As future work, we will consider linking more RS datasets with Freebase. We will also consider testing the performance of more knowledge-aware recommendation algorithms on more recommendation tasks using the linked dataset.

[2]Here, since our purpose is to illustrate the use of this linked dataset, we only select four methods for performance comparison. We will try more knowledge-ware recommendation algorithms as future work.

REFERENCES

[1] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *NIPS*. 2787–2795.

[2] Sarah Bouraga, Ivan Jureta, Stéphane Faulkner, and Caroline Herssens. 2014. Knowledge-Based Recommendation Systems: A Survey. *IJIIT* 10, 2 (2014), 1–19.

[3] Tianqi Chen, Weinan Zhang, Qiuxia Lu, Kailong Chen, Zhao Zheng, and Yong Yu. 2012. SVDFeature: a toolkit for feature-based collaborative filtering. *Journal of Machine Learning Research* (2012).

[4] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. 2015. Content-Aware Point of Interest Recommendation on Location-Based Social Networks. In *AAAI*. 1721–1727.

[5] Google. 2016. Freebase Data Dumps. https://developers.google.com/freebase/data.

[6] F. Maxwell Harper and Joseph A. Konstan. 2016. The MovieLens Datasets. *TiiS* 5, 4 (2016), 1–19.

[7] Ruining He, Wang-Cheng Kang, and Julian McAuley. 2018. Translation-based Recommendation: A Scalable Method for Modeling Sequential Behavior. In *IJCAI*.

[8] Ruining He and Julian Mcauley. 2016. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. In *WWW*.

[9] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S. Yu. 2018. Leveraging Meta-path based Context for Top- N Recommendation with A Neural Co-Attention Model. In *SIGKDD*. 1531–1540.

[10] Jin Huang, Wayne Xin Zhao, Hong-Jian Dou, Ji-Rong Wen, and Edward Y. Chang. 2018. Improving Sequential Recommendation with Knowledge-Enhanced Memory Networks. In *SIGIR*.

[11] Mohsen Jamali and Martin Ester. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In *RecSys*. 135–142.

[12] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* (2009), 30–37.

[13] Hao Ma, Irwin King, and Michael R. Lyu. 2009. Learning to recommend with social trust ensemble. In *SIGIR*. 203–210.

[14] Steffen Rendle. 2012. Factorization Machines with libFM. *ACM TIST* (2012).

[15] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI*.

[16] Markus Schedl. 2016. The LFM-1b Dataset for Music Retrieval and Recommendation. In *ICMR*.

[17] Tatjana Scheffler, Rafael Schirru, and Paul Lehmann. 2012. Matching Points of Interest from Different Social Networking Sites. In *KI*. 245–248.

[18] Chuan Shi, Chong Zhou, Xiangnan Kong, Philip S. Yu, Gang Liu, and Bai Wang. 2012. HeteRecom: a semantic-based recommendation systemin heterogeneous networks. In *SIGKDD*. 1552–1555.

[19] Yizhou Sun and Jiawei Han. 2012. Mining heterogeneous information networks: a structural analysis approach. *SIGKDD Explorations* 14, 2 (2012), 20–28.

[20] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. Ripple Network: Propagating User Preferences on the Knowledge Graph for Recommender Systems. (2018).

[21] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep Knowledge-Aware Network for News Recommendation. In *WWW*. 1835–1844.
[22] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE TKDE* (2017).
[23] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. 2014. Personalized entity recommendation: a heterogeneous information network approach. In *WSDM*. 283–292.
[24] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative Knowledge Base Embedding for Recommender Systems. In *SIGKDD*. 353–362.
[25] Wayne Xin Zhao, Yanwei Guo, Yulan He, Han Jiang, Yuexin Wu, and Xiaoming Li. 2014. We know what you want to buy: a demographic-based system for product recommendation on microblogs. In *KDD*.