The Effect of Semantic Knowledge Graph Richness on Embedding Based Recommender Systems

Daan L. Di Scala[1,†], Xu Wang[2,†] and Christopher Brewster[1,2]

[1]TNO Netherlands Organisation for Applied Scientific Research, Department Data Science, Kampweg 55, 3769 ZG Soesterberg, The Netherlands.

[2]Institute of Data Science, Maastricht University, Paul-Henri Spaaklaan 1, 6229 GT, Maastricht, The Netherlands

Abstract

Recommender systems, a specialised subfield within information retrieval, are crucial for identifying items that align with users' preferences. A knowledge graph-based recommender system can excel in the task of making recommendations due to the rich semantic information inherent in knowledge graphs. In this paper, our central focus is to investigate the impact of the semantic richness of knowledge graphs on the effectiveness of such recommender systems. To explore this research topic, we focus on the movie recommendation domain. For this, we create seven movie ontologies with varying levels of semantic richness, and combine these ontologies with movie data from the MovieLens 1M dataset and augmented using additional open linked data derived from Wikidata to produce seven different movie knowledge graphs. We provide the ontologies and knowledge graphs in an open-source repository. We then conduct experiments using two different approaches, consisting of nine Knowledge Graph Based Recommending (KGBR) methods and four Link Prediction (LP) methods based on Knowledge Graph Embeddings (KGE). The results demonstrate that richness of the knowledge graph does not impact the performance of KGBR methods significantly, but has a considerable impact on the KGE-based LP methods. We furthermore compare the best performing KGBR methods with the KGE-based LP methods, showing that the LP methods outperform all other recommendation methods when paired with the most extensive knowledge graph. From this, we conclude that the richness of the knowledge graph does not have a significant impact if the method already integrates other recommending approaches, but can heavily impact the LP methods employing the KGE approach, which interprets relationships as translations in embedding spaces. This supports the idea that using extended knowledge graphs is an effective approach for successful recommender systems.

Keywords

Recommender Systems, Ontologies, Knowledge Graphs, Embedding, Link Prediction

1. Introduction

Recommender systems play a critical role in a world where individuals are subject to more and more digital information and must make choices whether to purchase goods or access information. Recommendations can be applicable to many scenarios, such as suggesting which movie to watch, which book to read, which restaurant to go or which products to buy. The central purpose of recommender systems is to meet the need or preference of the people or group in question, therefore personalised decision support is needed for a successful recommendation system. The recommending process is implemented by algorithms able to analyse a user's data and profile, as well as the landscape of possibilities, and thus to predict items that the user might be interested in.

There are many types of recommender systems, whose algorithms are based on techniques such as content-based filtering, collaborative filtering, hybrid filtering, or popularity metrics, each with their own strengths [1, 2, 3]. Such algorithms can take advantage of Knowledge Graphs (KGs), which offer advantages, as these systems can leverage semantic knowledge and contextual information to provide more accurate, personalised and nuanced recommendations. Such recommendations based on semantics and context can be helpful for both users actively searching for items of their personal interest, as well

KaRS 2024: Knowledge-aware and Conversational Recommender Systems Workshop, October 14–18, 2024, Bari, Italy

† These authors contributed equally.

daan.discala@tno.nl (D. L. Di Scala); xu.wang@maastrichtuniversity.nl (X. Wang); christopher.brewster@tno.nl (C. Brewster)

0000-0003-1548-6675 (D. L. Di Scala); 0000-0002-7585-759X (X. Wang); 0000-0001-6594-9178 (C. Brewster)

as passive users that can gain from tailored results. Therefore, involving ontology-based knowledge graphs from diverse data sources can even further enhance the semantic value of these methods, leading to increasingly understandable recommendations. This research explores the influence of different types of knowledge graphs on the performance of recommender systems. Our focus here is to investigate the impact of the degree of semantic richness of knowledge graphs, where semantic richness is defined as the amount of additional formally represented data available for each item.

The main contributions of this paper are as follows: **1)** Seven modular movie ontologies with varying levels of semantic richness. **2)** Seven movie knowledge graphs based on the ontologies, combined with data from MovieLens 1M dataset, Wikidata and data produced from additional statistical analysis. **3)** An extensive comparison between nine Knowledge Graph Based Recommender Systems and four Link Prediction approaches. To measure the impact of the knowledge graph richness on the recommender system performance, we design an evaluation task using existing evaluation metrics. To support future reuse, we provide the ontologies, knowledge graphs, our code and dataset used in the experiments as open access in our repository[1].

The rest of this paper is structured as follows: in Section 2 we describe the background theory on recommendation systems, and more specifically, knowledge graph based recommender systems. In Section 3 we describe our method of ontology creation, knowledge graph generation and recommending model selection. In Section 4 we describe our experiments and in Section 5 we note our results and analysis. Finally, in Section 6 we conclude our paper and discuss possible future avenues.

2. Background

2.1. Recommender System Theory

Recommender systems are a type of information filtering systems designed to predict a user's ratings or preferences for items, thereby enhancing user experience and engagement. The fundamental function of these systems can be expressed as $\hat{r}_{ui} = f(u, i, p; \theta)$, where $\hat{r}_{ui}$ predicts the user u's interest in item i, p provides additional contextual parameters, and θ represents the model's parameters. These systems are categorised into: Content-Based Filtering, which recommends items similar to those a user has previously liked, based on item features ($\hat{r}_{ui} = \mathbf{x}_u^\top \mathbf{y}_i$), where $\mathbf{x}_u$ and $\mathbf{y}_i$ are feature vectors representing user preferences and item attributes, respectively; Collaborative Filtering, which makes recommendations using the ratings behaviour of other users ($\hat{r}_{ui} = \mathbf{p}_u^\top \mathbf{q}_i$), with $\mathbf{p}_u$ and $\mathbf{q}_i$ as the latent factor vectors for users and items, respectively; and Hybrid Approaches, combining both methods to leverage their strengths ($\hat{r}_{ui} = \alpha \cdot \mathbf{x}_u^\top \mathbf{y}_i + (1 - \alpha) \cdot \mathbf{p}_u^\top \mathbf{q}_i$), where α is a parameter balancing the contribution of each method. These methods allow recommender systems to tailor responses based on individual user preferences, thereby improving the efficiency and user satisfaction.

2.2. Knowledge Graph Based Recommender Systems

Ontologies formally represent concepts and relationships within a specific domain, which can be instantiated through Knowledge Graphs. Such knowledge graphs are used to represent a network of real-world entities — such as objects, events, situations, or concepts — and illustrates the interrelationships between them. It effectively organises information in a way that facilitates not only retrieval but also inference, making it an invaluable tool for enhancing the capabilities of recommender systems and other artificial intelligence applications. A knowledge graph can be modelled as a set of triples, denoted as (s, p, o), where s stands for the subject, p the predicate (or relation), and o the object. This triple format encapsulates the relationships within the data, allowing systems to leverage structured semantic information for advanced reasoning and query processing.

Knowledge Graph Based Recommender (KGBR) systems use such complex graph-structured networks of entities and their interrelationships to enhance recommendation accuracy and relevance to context

[1]https://github.com/XuWangDACS/Rich_KGRS

Figure 1: Ontology $\mathbf{O_0}$, the base movie ontology.

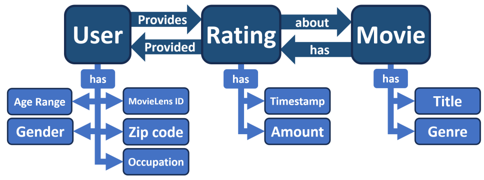


Figure 2: Ontology $\mathbf{O_1}$, extension is highlighted.

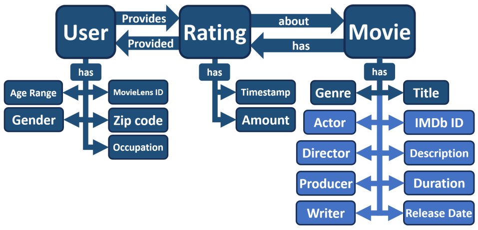


Figure 3: Ontology $\mathbf{O_2}$, extension is highlighted.

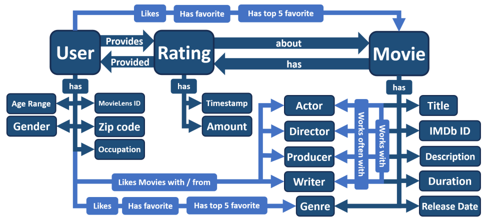


Figure 4: Ontology $\mathbf{O_3}$, extension is highlighted.

by leveraging semantically rich information inherent in the graphs. Wang et al. introduce the Ripp-MKR model, a novel multitask feature learning framework that leverages the benefits of knowledge graph embeddings through RippleNet to enhance recommender systems, demonstrating superior performance over existing methods in various recommendation scenarios [4]. Tu et al. introduce the Knowledge-aware Conditional Attention Networks (KCAN), a novel model that leverages knowledge graphs and conditional attention mechanisms to significantly enhance the accuracy and personalization of recommender systems [5]. Liu et al present presents CDKG-CE, a method for multi-domain item-item recommendation using cross-domain knowledge graph embedding, which addresses the issues of sparse data and cold-start in traditional recommender systems by efficiently linking items across various domains [6]. Another technique to improve recommendations is Link Prediction (LP). Knowledge Graph based Link Prediction can be used to anticipate potential links between entities in a knowledge graph based on the observed links and graph structure, which could be used to enhance the performance of these recommender systems.

3. Method

3.1. Dataset and Ontology Creation

To build the required knowledge graphs, we use three data sources: MovieLens 1M, Wikidata and additional statistical data. For the purpose of testing different levels of semantic richness, we create seven ontologies in RDF and RDFS, each with additional levels of information density and linking. These ontologies vary both in size as well as structure. Four ontologies, each designed with a gradually increasing level of semantic complexity, are set up as follows:

- $\mathbf{O_0}$ Base movie rating ontology. which includes concepts regarding users rating movies.
- $\mathbf{O_1}$ General Information Extension, which includes additional general attributes.
- $\mathbf{O_2}$ Movie Information Extension, which includes attributes specifically regarding the movies.
- $\mathbf{O_3}$ User-Movie Linking Extension, which includes links between Users and Movie attributes.

We create an additional 'plus' version of each ontology, which holds the extended attributions as concepts instead of literals, to support further extensions. These ontologies are called O_{1+}, O_{2+} and O_{3+}. All of the ontologies (with the namespace MR: <HTTP://ONTOLOGY.TNO.NL/MOVIE-RATING-ONTOLOGY/>) are included in our repository in the TTL file format[1].

Ontology O_0 (Figure 1) serves as the base ontology, which holds just three concepts for a User, which provides a Rating about a Movie. These concepts are linked with each other through the MR:HASRATING, MR:ISABOUTMOVIE, MR:PROVIDESRATING, and MR:PROVIDEDBY relations.

Continuing on top of O_0, ontologies O_1 and O_{1+} (Figure 2) serve as extensions that also include semantic properties regarding MovieLens. They include concepts that belong to the Users: MR:GENDER, MR:ZIPCODE, MR:MOVIELENSID, MR:OCCUPATION, concepts that belong to the Ratings: MR:TIMESTAMP and MR:AMOUNTOFSTARS, as well as concepts that belong to the Movies: MR:TITLE and MR:GENRE.

On top of O_{1+}, we build ontologies O_2 and O_{2+} (Figure 3). These extensions also contain information from IMDb through Wikidata. They include concepts for Movies' MR:IMDBID, MR:DURATION, MR:RELEASEDATE, and associated parties: MR:ACTOR, MR:PRODUCER, MR:DIRECTOR and MR:WRITER.

Finally, on top of O_{2+}, we build ontologies O_3 and O_{3+} (Figure 4). These extensions also convey extended information based on statistical information about the Users and the Movies they have rated. They include concepts for a User Liking Movies and Genres (defined by having the Rating >2.5), Having Favourite Movies and Genres (defined by highest Rating) and Having Top 5 Favourite Movies and Genres (defined by top 5 defined by highest Rating). Furthermore, it includes relations when the User Likes movies that involve certain Actors, Directors, Producers and Writers (defined by having the Rating of a Movie with these Persons involved >2.5). Finally, it includes additional links between Actors, Directors, Producers and Writers that have worked together with each other, and even works often with each other (defined by having an above average cooperation rate over all of the movies).

3.2. Knowledge Graph Generation

Based on the ontology concepts and relations we have previously defined, we generate a movie-user knowledge graph by integrating the MovieLens datasets with Wikidata's open data and mapping this data to the ontologies. The initial step involves mapping the MovieLens datasets to Wikidata. The MovieLens datasets, specifically the 1M dataset used in this research, provide basic information about users, movies, and user interactions. However, these datasets lack semantic depth. To enhance this, we extract additional movie data from Wikidata. Notably, the MovieLens 1M dataset can be easily linked to the MovieLens 20M dataset as they share the same MovieLens movie IDs. The 20M dataset includes links to IMDb IDs, which are widely used in IMDb's non-commercial data[2]. Given that movie entities in Wikidata are also linked to these IMDb IDs, we can map movies from the MovieLens 1M dataset to corresponding entries in Wikidata using IMDb IDs.

After establishing the necessary mappings, we enrich the knowledge graph with detailed movie data retrieved from Wikidata that is not available in the MovieLens datasets. In constructing the knowledge graph, we use predicate URIs that are not only defined in our custom ontology but also those mapped from Wikidata. The entity URIs within the knowledge graph leverage Wikidata URIs for the data sourced therefrom, and we create additional URIs consistent with the namespace defined in our ontology.

3.3. Recommending Methods

We compare the following nine KGBR systems: CKE integrates collaborative filtering and knowledge base embeddings to enrich the feature representation of users and items, leveraging both user-item interactions and semantic knowledge graph data [7]. CFKG enhances collaborative filtering methods by incorporating knowledge graph data to improve recommendations, especially in cases of sparse user-item interactions [8]. KGAT applies an attention-based graph neural network to exploit the hierarchical structure of knowledge graphs for recommendation, enhancing the interpretability of the relationships between users and items [9]. KGCN leverages graph convolutional networks to extract high-level features from knowledge graphs, enabling a more accurate and context-aware recommendation system [10]. KGIN incorporates additional informative nodes into graph networks, using deep semantic relationships within knowledge graphs to boost recommendation accuracy [11]. KGNNLS combines knowledge graph neural networks with label smoothness regularization to smooth the learning process

[2]https://developer.IMDb.com/non-commercial-datasets/

and enhance the quality of recommendations [12]. KTUP employs tensor factorization techniques to unify predictions across users, items, and knowledge graph entities, enhancing the model's predictive capabilities [13]. MKR is a multi-task learning framework that enhances recommendation tasks by simultaneously predicting knowledge graph linkages, leveraging shared feature learning between tasks [14]. RippleNet simulates the ripple effect across knowledge graphs by propagating user preferences over the set relations in a graph, dynamically updating recommendations based on these interactions [15].

In addition to these nine methods, we examine the impact of semantic richness on the effectiveness of recommendation systems by framing the recommendation task as a link prediction challenge. The goal is to predict potential recommendation links between users and movies. For this purpose, we employ the TransE [16], DistMult [17], RotatE [18], and SimplE [19] models, to learn vector representations of entities and relations to capture and infer relationships within a knowledge graph, each using different mathematical approaches to model these interactions.

4. Experiments

We run 91 experiments (13 recommender methods on seven knowledge graphs) to research the impact of richness of knowledge graph on the effectiveness of recommender systems. These experiments are marked as $\mathbf{KG_0}$ through $\mathbf{KG_{3+}}$, as shown in Table 1. The source code and datasets of our experiments can be found at our repository[1].

We use the RecBole [20] Python library in our experiments. RecBole provides many existing recommender system implementations, including the knowledge based methods we evaluate in our experiments. We use the seven generated knowledge graphs (based on mapping the data sources on the created ontologies) as discussed in Section 3 as the knowledge background for RecBole's knowledge based methods. We use the PyKEEN [21] Python library to train the TransE embedding of the knowledge graph. After training, we use the cosine similarity between embeddings of user and movie to predict the potential link between them.

Four metrics are used to evaluate the performance of recommendation: **MRR@k** (Mean Reciprocal Rank at k): Computes the average of reciprocal ranks of the first relevant item in the top-k results, emphasising the importance of higher-ranked relevant items. **NDCG@k** (Normalised Discounted Cumulative Gain at k): Evaluates the ranking quality of the top-k recommendations by considering the position of relevant items and applying a logarithmic discount based on rank [22]. **Hit@k** (Hit Rate at k): Calculates the proportion of queries for which at least one relevant item appears among the top-k recommendations, effectively measuring the model's ability to retrieve relevant items within the top results. And finally, **Precision@k**: Assesses the fraction of relevant items among the top-k recommendations made by a model. For RecBole, these metrics are already included. For the Link Prediction based methods, we use the ranx [23] Python library to get these measures. In this paper, we choose to set the k of these metrics as 10. This provides a good indication, because the top 10 recommendations is usually of the most importance. The top 10 recommended items is likely to be the most visible to users, especially on platforms where the first page or screen carries high importance.

5. Results and Discussion

5.1. Results

Table 1 shows the results of the evaluation metrics of all recommender methods over the seven experiment settings (KG_1 until KG_{3+}). Highest scores are marked with bold text, second highest are underlined, and each of the metrics the best performing KGBR and LP method are highlighted. As evident from the Table, the best performing methods are KGIN for KGBR methods and and SimplE for LP methods, with SimplE gaining a significant impact on KG_{3+}.

By analysing Table 1, we can conclude the following: 1) For KGBR methods, the improvement on semantic richness of knowledge graph does not significantly impact the performance of recommendation,

Table 1
Evaluation Metrics Results of Recommender Systems

Method	KG$_0$	KG$_1$	KG$_{1+}$	KG$_2$	KG$_{2+}$	KG$_3$	KG$_{3+}$
Hit @ 10							
CKE	0.737	0.735	0.725	0.725	0.734	0.734	0.734
CFKG	0.722	0.730	0.734	0.722	0.733	0.728	0.728
KGAT	0.730	0.722	0.719	0.720	0.721	0.726	0.710
KGCN	0.705	0.713	0.705	0.706	0.697	0.683	0.685
KGIN	**0.754**	**0.760**	**0.758**	**0.757**	**0.753**	0.758	0.755
KGNN	0.705	0.713	0.705	0.706	0.697	0.683	0.686
KTUP	0.627	0.590	0.591	0.596	0.583	0.562	0.559
MKR	0.699	0.697	0.697	0.695	0.699	0.698	0.691
RNet	0.640	0.642	0.651	0.654	0.651	0.643	0.642
LPDMult	0.018	0.165	0.097	0.000	0.104	**0.835**	0.922
LPRotatE	0.394	0.288	0.389	0.424	0.504	0.373	0.209
LPSimplE	0.271	0.065	0.002	0.000	0.058	0.587	**0.933**
LPTransE	0.126	0.099	0.135	0.091	0.112	0.176	0.817
Precision @ 10							
CKE	**0.198**	**0.198**	0.196	0.198	0.196	0.198	0.198
CFKG	0.195	0.197	0.196	0.196	0.196	0.194	0.195
KGAT	0.193	0.192	0.190	0.190	0.193	0.191	0.186
KGCN	0.186	0.187	0.184	0.187	0.184	0.177	0.179
KGIN	**0.198**	**0.198**	**0.202**	**0.202**	**0.200**	0.200	0.200
KGNN	0.186	0.186	0.184	0.187	0.184	0.177	0.179
KTUP	0.154	0.140	0.139	0.142	0.138	0.128	0.129
MKR	0.183	0.183	0.180	0.181	0.184	0.184	0.178
RNet	0.159	0.159	0.166	0.158	0.161	0.159	0.157
LPDMult	0.005	0.101	0.019	0.000	0.026	**0.368**	0.437
LPRotatE	0.109	0.071	0.105	0.139	0.147	0.094	0.042
LPSimplE	0.065	0.024	0.000	0.000	0.007	0.290	**0.443**
LPTransE	0.015	0.010	0.018	0.018	0.016	0.024	0.362

Method	KG$_0$	KG$_1$	KG$_{1+}$	KG$_2$	KG$_{2+}$	KG$_3$	KG$_{3+}$
Hit @ 10							
CKE	0.737	0.735	0.725	0.725	0.734	0.734	0.734
CFKG	0.722	0.730	0.734	0.722	0.733	0.728	0.728
KGAT	0.730	0.722	0.719	0.720	0.721	0.726	0.710
KGCN	0.705	0.713	0.705	0.706	0.697	0.683	0.685
KGIN	**0.754**	**0.760**	**0.758**	**0.757**	**0.753**	0.758	0.755
KGNN	0.705	0.713	0.705	0.706	0.697	0.683	0.686
KTUP	0.627	0.590	0.591	0.596	0.583	0.562	0.559
MKR	0.699	0.697	0.697	0.695	0.699	0.698	0.691
RNet	0.640	0.642	0.651	0.654	0.651	0.643	0.642
LPDMult	0.018	0.165	0.097	0.000	0.104	**0.835**	0.922
LPRotatE	0.394	0.288	0.389	0.424	0.504	0.373	0.209
LPSimplE	0.271	0.065	0.002	0.000	0.058	0.587	**0.933**
LPTransE	0.126	0.099	0.135	0.091	0.112	0.176	0.817
Precision @ 10							
CKE	**0.198**	**0.198**	0.196	0.198	0.196	0.198	0.198
CFKG	0.195	0.197	0.196	0.196	0.196	0.194	0.195
KGAT	0.193	0.192	0.190	0.190	0.193	0.191	0.186
KGCN	0.186	0.187	0.184	0.187	0.184	0.177	0.179
KGIN	**0.198**	**0.198**	**0.202**	**0.202**	**0.200**	0.200	0.200
KGNN	0.186	0.186	0.184	0.187	0.184	0.177	0.179
KTUP	0.154	0.140	0.139	0.142	0.138	0.128	0.129
MKR	0.183	0.183	0.180	0.181	0.184	0.184	0.178
RNet	0.159	0.159	0.166	0.158	0.161	0.159	0.157
LPDMult	0.005	0.101	0.019	0.000	0.026	**0.368**	0.437
LPRotatE	0.109	0.071	0.105	0.139	0.147	0.094	0.042
LPSimplE	0.065	0.024	0.000	0.000	0.007	0.290	**0.443**
LPTransE	0.015	0.010	0.018	0.018	0.016	0.024	0.362

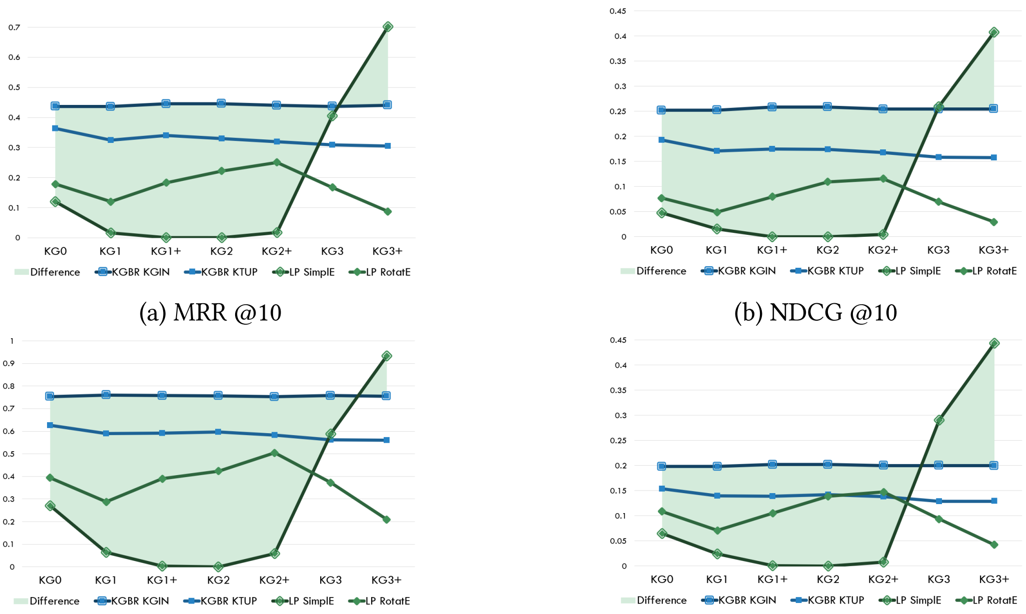
but a small improvement can be seen with CFKG, KGIN and RippleNet. 2) For the Link Prediction recommending methods, semantic richness of knowledge graph does show a remarkable impact on the performance of recommendation, especially on SimplE, DistMult and TransE. 3) With the first knowledge graphs KG$_0$ and KG$_1$, CKE performs best, while with the knowledge graphs KG$_{1+}$ and KG$_2$ KGIN performs best. On KG$_{2+}$ DistMult and on SimplE outperforms all other methods on all metrics on the final experiment, which uses KG$_{3+}$, the knowledge graph with the highest semantic richness. Therefore, on the first knowledge graphs with less semantic richness, KGBR methods perform better, while LP methods thrive based on richer knowledge graphs.

Figure 5 shows the comparison between best and worst performing KGBR and LP methods. Notably, the KGBR methods' performance remain stable through all different experiments. However, the LP method RotatE is affected positively up until KG$_{2+}$. After KG$_{2+}$, its performance drops again. The best LP method however, SimplE, follows a similar pattern as DistMult and TransE, and only significantly improves after introducing more links from KG$_{2+}$ on. With the introduction of more semantic links in the knowledge graphs, SimplE surpasses the KGBR methods in performance with KG$_3$ on Precision and NDCG, and with KG$_{3+}$ on Hit and MRR.

5.2. Discussion

Regarding the dataset, while MovieLens is a hugely useful because of its user data and the most popular dataset in recent research [24], it is released in 2003 and therefore does not take current movies into account. As movie recommendation is seen in popular everyday household streaming services such as Netflix [25, 26], it is interesting to further extend this research with more current data. Our focus on only the MovieLens dataset is a limitation, which could be expanded upon. Popular social platforms where users track their favourite items, such as Letterboxd[3] for movies or Goodreads[4] [27] for books, could be examined for this, and can provide more recent information. Furthermore, it would be possible

[3]https://letterboxd.com/
[4]https://www.goodreads.com/



(a) MRR @10

(b) NDCG @10

(c) Hit @10

(d) Precision @10

Figure 5: Performance comparison of best (KGIN) and worst (KTUP) performing KGBR methods, and best (SimplE) and worst (RotatE) performing LP methods. The difference between best KGBR and best LP results is marked as colored area between the graph lines.

to enrich our generated knowledge graphs with other data sources through Wikidata, as Wikidata links movie entities to many other data sources.

One limitation of this work is that both the ontology creation and knowledge graph mapping is a manual, intricate process. However, automatically providing such mappings is still the subject of ongoing research in the field of Ontology Mapping [28, 29]. To ensure the ontologies were focused, we purposefully did not model the entire movie domain. To ensure relevancy, concepts and relations that we did include are based on MovieLens and Wikidata, but the ontologies (as provided in our repository[1]) could be expanded upon with further concepts and relations.

Furthermore, it is only when we introduce additional links in the transition to ontology O_{3+} that there is a noticeable impact in LP performance. This might be the reason why the Link Prediction methods have such a large increase in performance in KG_{3+}. Future research could be done into extending the ontology and subsequent knowledge graph even more with additional information, with a specific focus on the amount of semantic links to see which parts of ontology extensions have the most impact on recommendations. Furthermore, our research focuses on the movie domain as popular example, but further steps would need to be taken to generalise this to other domains.

Regarding the recommendation methods, as the LP methods show the most promising outcomes, it would be a valuable future step to research other LP approaches more deeply. Other LP methods, such as path based or neighbourhood based methods could therefore provide further insight in the effect of this ontology based approach.

Regarding the results, we find that the performance of the KGBR methods stayed relatively unchanged without being affected by the change of semantic richness of the knowledge graph. This stability is indicative that KGBR methods maintain consistent performance, irrespective of varying knowledge graph richness. Notably, most methods do not exhibit a consistent trend of improvement or deterioration, suggesting that the modifications introduced in each experiment have a balanced impact on performance.

KTUP shows a significant decline in all metrics from KG_0 to KG_{3+}, indicating potential issues with adapting to changes over the experiments. SimplE, in contrast, exhibits dramatic fluctuations, especially pronounced on KG_{3+}. Given that SimplE primarily relies on embedding similarities, the enriched semantic content of the knowledge graph can directly influence its embeddings, enhancing its performance. Furthermore, KGIN consistently shows the best performance among the KGBR methods we evaluated. This underscores its effectiveness in leveraging the features of the knowledge graph across diverse experimental setups.

Most LP methods outperformed KGBR methods on all metrics on KG_{3+}, based on Table 1. For the MRR score, they achieve more than 0.55, which means that these recommender systems could relative-quickly surface the most relevant items. Their NDCG scores reach 0.3, which means that they cannot correctly put the most relevant results on the top-ranked items of the recommendation list. Hit and Precision scores reach 0.8 and 0.35, respectively, which means that they can always recommend at least one relevant result on top-10 list (based on Hit@10) and can recommend more than three relevant results on this list (based on Precision@10). The similar performance of DistMult, SimplE, and TransE in link prediction tasks comes from their reliance on linear operations and their ability to model simple relationships. These models perform well on datasets where these assumptions are valid, leading to comparable results, specifically for KG_{3+}. However, the RotatE method underperformed. RotatE is designed to model complex relational structures using rotations in a complex vector space, which allows it to handle more intricate and non-linear relationships better than the simpler models. RotatE's ability ability to manage a wider range of relational patterns, makes it a stronger choice for knowledge graphs with more complex and diverse relationships. Methods like KGIN and RippleNet show very marginal changes, indicating stability in their performance despite the enhanced richness of KG_{3+}. This suggests that simply adding more semantic richness to a knowledge graph does not universally translate to improved recommendation performance.

6. Conclusion & Future Work

In this paper, we explored the impact of the semantic richness of a knowledge graph for knowledge based recommending methods. To test this, we focused on the movie recommendation domain. We created seven movie ontologies which are used to build seven knowledge graphs with varying semantic richness and linking. We enriched the ontology-based knowledge graphs with MovieLens 1M data, information from wikidata and supplemental statistical data. We then designed and conducted 91 experiments, and compared the performance of state-of-the-art knowledge based recommending methods and link prediction methods on four existing ranking evaluation metrics. The results indicate that increasing the semantic richness of the knowledge graph does not bring significant impact on the knowledge based recommending methods, but does significantly impact the link prediction approaches.

This paper built its enrichment approach using manually created movie ontologies. However, future research could explore the potential advantages of a learning based approach to ontology enrichment, to further tailor the recommendations to the specific needs of the user. By using machine learning algorithms, researchers could study the process of automated adaption or enhancement of such ontologies and subsequent knowledge graphs. This shift could lead to more dynamic and comprehensive recommender systems, as learning-based systems are capable of continuously updating and refining their knowledge base from vast amounts of data. Finally, increasing the semantic strength can impact the understandability of the provided recommendendations. This paper does not further consider the impact of semantic richness of knowledge graphs on explainable recommender systems, particularly knowledge based or graph based explainable recommender systems [30, 31, 32]. Increasing semantic richness of a knowledge graph could also increase more paths between users and movies in the knowledge graph, which could bring more possible explanations in the recommendations. More research into combining extensive semantic knowledge graphs and explainable recommender systems could therefore be useful to even further strengthen the recommender systems.

Acknowledgements

This work is funded by the HORIZON EUROPE project "EU-FarmBook: supporting knowledge exchange between all AKIS actors in the European Union" (Grant ID: 101060382).

References

[1] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, M. Wang, Lightgcn: Simplifying and powering graph convolution network for recommendation, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '20, ACM, 2020. URL: http://dx.doi.org/10.1145/3397271.3401063. doi:10.1145/3397271.3401063.

[2] X. Wang, X. He, M. Wang, F. Feng, T.-S. Chua, Neural graph collaborative filtering, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '19, ACM, 2019. URL: http://dx.doi.org/10.1145/3331184.3331267. doi:10.1145/3331184.3331267.

[3] H. Guo, R. Tang, Y. Ye, Z. Li, X. He, Deepfm: a factorization-machine based neural network for ctr prediction, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI'17, AAAI Press, 2017, p. 1725–1731.

[4] Y. Wang, L. Dong, Y. Li, H. Zhang, Multitask feature learning approach for knowledge graph enhanced recommendations with ripplenet, PLOS ONE 16 (2021) e0251162. URL: http://dx.doi.org/10.1371/journal.pone.0251162. doi:10.1371/journal.pone.0251162.

[5] K. Tu, P. Cui, D. Wang, Z. Zhang, J. Zhou, Y. Qi, W. Zhu, Conditional graph attention networks for distilling and refining knowledge graphs in recommendation, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, CIKM '21, ACM, 2021. URL: http://dx.doi.org/10.1145/3459637.3482331. doi:10.1145/3459637.3482331.

[6] J. Liu, W. Huang, T. Li, S. Ji, J. Zhang, Cross-domain knowledge graph chiasmal embedding for multi-domain item-item recommendation, IEEE Transactions on Knowledge and Data Engineering (2022) 1–1. URL: http://dx.doi.org/10.1109/TKDE.2022.3151986. doi:10.1109/tkde.2022.3151986.

[7] F. Zhang, N. J. Yuan, D. Lian, X. Xie, 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, KDD '16, ACM, 2016. URL: http://dx.doi.org/10.1145/2939672.2939673. doi:10.1145/2939672.2939673.

[8] Q. Ai, V. Azizi, X. Chen, Y. Zhang, Learning heterogeneous knowledge base embeddings for explainable recommendation, Algorithms 11 (2018) 137. URL: http://dx.doi.org/10.3390/a11090137. doi:10.3390/a11090137.

[9] X. Wang, X. He, Y. Cao, M. Liu, T.-S. Chua, Kgat: Knowledge graph attention network for recommendation, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '19, ACM, 2019. URL: http://dx.doi.org/10.1145/3292500.3330989. doi:10.1145/3292500.3330989.

[10] H. Wang, M. Zhao, X. Xie, W. Li, M. Guo, Knowledge graph convolutional networks for recommender systems, in: The World Wide Web Conference, WWW '19, ACM, 2019. URL: http://dx.doi.org/10.1145/3308558.3313417. doi:10.1145/3308558.3313417.

[11] X. Wang, T. Huang, D. Wang, Y. Yuan, Z. Liu, X. He, T.-S. Chua, Learning intents behind interactions with knowledge graph for recommendation, in: Proceedings of the Web Conference 2021, WWW '21, ACM, 2021. URL: http://dx.doi.org/10.1145/3442381.3450133. doi:10.1145/3442381.3450133.

[12] H. Wang, F. Zhang, M. Zhang, J. Leskovec, M. Zhao, W. Li, Z. Wang, Knowledge-aware graph neural networks with label smoothness regularization for recommender systems, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '19, ACM, 2019. URL: http://dx.doi.org/10.1145/3292500.3330836. doi:10.1145/3292500.3330836.

[13] Y. Cao, X. Wang, X. He, Z. Hu, T.-S. Chua, Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences, in: The World Wide Web Conference, WWW '19, ACM, 2019. URL: http://dx.doi.org/10.1145/3308558.3313705. doi:10.1145/3308558.3313705.

[14] H. Wang, F. Zhang, M. Zhao, W. Li, X. Xie, M. Guo, Multi-task feature learning for knowledge graph enhanced recommendation, in: The World Wide Web Conference, WWW '19, ACM, 2019. URL: http://dx.doi.org/10.1145/3308558.3313411. doi:10.1145/3308558.3313411.

[15] H. Wang, F. Zhang, J. Wang, M. Zhao, W. Li, X. Xie, M. Guo, 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 '18, ACM, 2018. URL: http://dx.doi.org/10.1145/3269206.3271739. doi:10.1145/3269206.3271739.
[16] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in: C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K. Weinberger (Eds.), Advances in Neural Information Processing Systems, volume 26, Curran Associates, Inc., 2013. URL: https://proceedings.neurips.cc/paper_files/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf.
[17] B. Yang, W.-t. Yih, X. He, J. Gao, L. Deng, Embedding entities and relations for learning and inference in knowledge bases, 2014. URL: https://arxiv.org/abs/1412.6575. doi:10.48550/ARXIV.1412.6575.
[18] Z. Sun, Z.-H. Deng, J.-Y. Nie, J. Tang, Rotate: Knowledge graph embedding by relational rotation in complex space, 2019. URL: https://arxiv.org/abs/1902.10197. doi:10.48550/ARXIV.1902.10197.
[19] S. M. Kazemi, D. Poole, Simple embedding for link prediction in knowledge graphs, 2018.
[20] L. Xu, Z. Tian, G. Zhang, J. Zhang, L. Wang, B. Zheng, Y. Li, J. Tang, Z. Zhang, Y. Hou, X. Pan, W. X. Zhao, X. Chen, J. Wen, Towards a more user-friendly and easy-to-use benchmark library for recommender systems, in: SIGIR, ACM, 2023, pp. 2837–2847.
[21] M. Ali, M. Berrendorf, C. T. Hoyt, L. Vermue, S. Sharifzadeh, V. Tresp, J. Lehmann, PyKEEN 1.0: A Python Library for Training and Evaluating Knowledge Graph Embeddings, Journal of Machine Learning Research 22 (2021) 1–6. URL: http://jmlr.org/papers/v22/20-825.html.
[22] K. Järvelin, J. Kekäläinen, Cumulated gain-based evaluation of IR techniques, ACM Trans. Inf. Syst. 20 (2002) 422–446.
[23] E. Bassani, ranx: A blazing-fast python library for ranking evaluation and comparison, in: ECIR (2), volume 13186 of *Lecture Notes in Computer Science*, Springer, 2022, pp. 259–264. doi:10.1007/978-3-030-99739-7_30.
[24] Z. Sun, D. Yu, H. Fang, J. Yang, X. Qu, J. Zhang, C. Geng, Are we evaluating rigorously? benchmarking recommendation for reproducible evaluation and fair comparison, in: Proceedings of the 14th ACM Conference on Recommender Systems, 2020, pp. 23–32.
[25] C. A. Gomez-Uribe, N. Hunt, The netflix recommender system: Algorithms, business value, and innovation, ACM Transactions on Management Information Systems (TMIS) 6 (2015) 1–19.
[26] M. Jenner, Netflix and the Re-invention of Television, Springer, 2018.
[27] M. Thelwall, K. Kousha, Goodreads: A social network site for book readers, Journal of the Association for Information Science and Technology 68 (2017) 972–983.
[28] Y. He, J. Chen, D. Antonyrajah, I. Horrocks, Bertmap: a bert-based ontology alignment system, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 36, 2022, pp. 5684–5691.
[29] Y. Kalfoglou, M. Schorlemmer, Ontology mapping: the state of the art, The knowledge engineering review 18 (2003) 1–31.
[30] X. Wang, D. Wang, C. Xu, X. He, Y. Cao, T.-S. Chua, Explainable reasoning over knowledge graphs for recommendation, Proceedings of the AAAI Conference on Artificial Intelligence 33 (2019) 5329–5336. URL: http://dx.doi.org/10.1609/aaai.v33i01.33015329. doi:10.1609/aaai.v33i01.33015329.
[31] Z. Kurt, T. Köllmer, P. Aichroth, An explainable knowledge graph-based news recommendation system, in: Proceedings of the 15th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, SCITEPRESS - Science and Technology Publications, 2023. URL: http://dx.doi.org/10.5220/0012161300003598. doi:10.5220/0012161300003598.
[32] L. Xie, Y. Jin, Z. Cui, L. Wang, Research on explainable recommendation model based on knowledge graph, in: 2022 IEEE International Conference on Networking, Sensing and Control (ICNSC), IEEE, 2022. URL: http://dx.doi.org/10.1109/icnsc55942.2022.10004187. doi:10.1109/icnsc55942.2022.10004187.