MindReader: Recommendation over Knowledge Graph Entities with Explicit User Ratings

Anders H. Brams, Anders L. Jakobsen, Theis E. Jendal, Matteo Lissandrini, Peter Dolog, Katja Hose
{ahbr,alja,tjendal,matteo,dolog,khose}@cs.aau.dk
Department of Computer Science
Aalborg University

ABSTRACT

Knowledge Graphs (KGs) have been integrated in several models of recommendation to augment the informational value of an item by means of its related entities in the graph. Yet, existing datasets only provide explicit ratings on items and no information is provided about users' opinions of other (non-recommendable) entities. To overcome this limitation, we introduce a new dataset, called the MindReader dataset, providing explicit user ratings both for items and for KG entities. In this first version, the MindReader dataset provides more than 102 thousands explicit ratings collected from 1,174 real users on both items and entities from a KG in the movie domain. This dataset has been collected through an online interview application that we also release as open source. As a demonstration of the importance of this new dataset, we present a comparative study of the effect of the inclusion of ratings on non-item KG entities in a variety of state-of-the-art recommendation models. In particular, we show that most models, whether designed specifically for graph data or not, see improvements in recommendation quality when trained on explicit non-item ratings. Moreover, for some models, we show that non-item ratings can effectively replace item ratings without loss of recommendation quality. This finding, in addition to an observed greater familiarity from users towards certain descriptive entities than movies, motivates the use of KG entities for both warm and cold-start recommendations.

ACM Reference Format:
Anders H. Brams, Anders L. Jakobsen, Theis E. Jendal, Matteo Lissandrini, Peter Dolog, Katja Hose. 2020. MindReader: Recommendation over Knowledge Graph Entities with Explicit User Ratings. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland*. ACM, New York, NY, USA, 8 pages.
https://doi.org/10.1145/3340531.3412759

1 INTRODUCTION

The goal of Recommender Systems (RSs) is to recommend items (e.g., products) to users based on some understanding of both the item attributes and the users' preferences for them [1]. They have been widely adopted in many online systems [1, 3]. In order to infer the preferences of a user, RSs require at least some information about past user-item interactions, which is usually referred to as user feedback (explicit or implicit). Explicit feedback comes from interactions where the user's intention is to provide feedback (e.g., an explicit rating) [3, 10, 12, 19, 30]. Implicit feedback is collected when it is not the user's intention to actually provide any explicit judgement (e.g., a product purchase) [15]. To enrich the available information, e.g., with additional item features, the system can also employ a KG [2, 5]. A KG is a heterogeneous graph representing entities like products, people, places, and concepts as nodes, and the relationships among them as typed edges. Hence, this unifying model combines both recommendable items (the products) and all other non-recommendable items (called descriptive entities). It has been shown that it is possible to leverage explicit feedback from the user for some descriptive entities [8]. In particular, prior works [11, 27] have sought to investigate the effects of incorporating ratings on item tags finding that both inferred and explicit ratings on tags lead to higher recommendation quality. However, tags (i) have no semantic inter-relations, (ii) have no explicit semantic relationship to items, and (iii) vary greatly in quality [11].

Instead, in this work, we address user ratings on descriptive entities from a heterogeneous KGs, overcoming the aforementioned issues (Section 2). To the best of our knowledge, no prior study has investigated the effect of including user explicit feedback over non-recommendable entities in a KG (Section 3). Moreover, no public dataset exists with this kind of information. Therefore, the first contribution of this work is the MindReader dataset (publicly available at https://mindreader.tech/dataset). This dataset has been collected through a data collection platform that asks real users to provide ratings on both recommendable and descriptive entities in the movie domain (Section 4). Hence, as a second contribution, we also release an extensible open-source platform for collecting user rating for KG-enhanced datasets. The analysis of this data (Section 5) provides a number of insights w.r.t. how user preferences correlate to various types of KG entities. Finally, to demonstrate whether user ratings on descriptive entities can be beneficial in generating more personalised recommendations, we investigate the effect of their inclusion in a large set of state-of-the-art machine learning models for personalised recommendation (Section 6). Thanks to this new dataset, as a third contribution, we provide some initial findings that explain how the inclusion of ratings on descriptive entities affects the quality of recommendations. Among others, our results suggest that descriptive entity ratings can serve as replacements for recommendable entity ratings in the warm-start setting, and motivate their utility also in the cold-start setting.

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3412759

2 KG-GUIDED RECOMMENDATIONS

Consider the case of movie recommendations. When modelling user preferences, we are concerned with what type of content users prefer, for instance, whether they like Science Fiction, dislike Horror, or prefer movies by some specific director. Some of these preferences can be combined in different ways, e.g., a user could generally like biographies but generally dislike science fiction, as shown in Figure 1. Yet, in inferring such complex preferences, observations on recommendable entities alone can be insufficient. Additionally, to infer user preferences from a small set of user-entity observations, eliciting feedback towards descriptive entities (e.g., actors and genres instead of movies) can intuitively be more informative. Finally, actors and genres have drastically different relations to movies, and hence a user's preferences towards either will affect the user's movie preferences differently. For example, a user disliking "Cloud Atlas" may lead us to inferring that the user dislikes science fiction and movies starring Tom Hanks, preventing us from recommending "Catch Me If You Can" although the user likes biographies. Without an appropriate modelling we cannot imagine to infer all these nuances. Moreover, having access to explicit user feedback on descriptive entities is particularly important for the cold-start settings [6, 25]. In a cold-start scenario, we are provided with a set of user preferences for items and a new unseen item is added to the database. In this case, it is possible to transfer the information from descriptive entities to the new unseen item. The complementary cold-start setting is a new user accessing the system. In this setting, a common strategy is instead to conduct an interview with the user to determine their preferences [6, 25]. If one were to most quickly determine a user's movie preferences from a blank slate, it makes little sense to immediately ask towards a specific set of movies [6, 25]. Among others, there is the possibility that the user might not be familiar with the movies they are asked about. Instead, since users generally have an opinion about Horror movies, even when not familiar with many movies of that genre, asking about their preference towards the genre is more likely to provide reliable information. Therefore, in this work we provide the first dataset with user-entity ratings both for recommendable and descriptive entities. This will allow the study of methodologies to develop novel KG-guided recommendation systems to address all the issues mentioned above.

3 RELATED WORK

A multitude of datasets for personalised recommendation exists [3, 12, 18, 33], and several works have considered using KGs for preference elicitation [2, 5]. Common for all these existing datasets is that they only contain explicit user ratings on recommendable entities. Few works have explored the advantages of eliciting explicit user feedback on objects constituting non-recommendable items, in particular product tags [11, 27] and item features [8, 20, 25]. Yet, none of the existing datasets provide ratings for KG entities that are non-recommendable items.

Ratings for tags. *Tagommenders* [27] provides ratings by 995 MovieLens users for 118, 017 movie tag. Here, tags comprise a set of labels that are generally ascribed to genres and categories (e.g., "french movies" or "terrorism"). Hence, the effects of tag-based RSs are evaluated in comparison to recommendable-entity-based RSs, finding that tag-based RSs using explicit tag ratings outperform recommendable-entity-based RSs that rely on Collaborative Filtering (CF) methods. Furthermore, a linear combination of the best tag-based RS and the best recommendable-entity-based RS is demonstrated to yield the best results. Another work proposes *regress-tag* [11] collecting explicit user ratings for specific movie-tag pairs over ∼100 movies and 19 users, for 5, 648 ratings in total. Thus, this dataset collects multiple ratings from the same user for the same tag, depending on the movie to which it is assigned. The results show that using explicitly collected tag ratings provides better performance than implicitly derived tag ratings. Finally, *ShoppingAdvisor* [8] collects two datasets of explicit and implicit user feedback on cars and cameras, respectively. In practice, keywords appearing in reviews and product tags are extracted and used to describe user-item relationships. This association is inferred as a positive relationship. For instance, a camera can be tagged as "food" and "nature" if a given user has used those tags on an picture taken with the camera, and should be recommended to a user looking for a camera to take pictures of food or nature. As for the ground-truth ranking data, popularity is used as a proxy for ground-truth ratings, ranking the cameras by the number of pictures taken with a given camera for every single tag. Although the ShoppingAdvisor datasets contain a large variety of such tags on the recommendable entities [8], they suffer from the same limitations as Tagommenders [27] as the tags are free-text and added by users without restriction.

All the above works differ to MindReader because the ratings are collected on user-provided tags rather than on entities in a reference knowledge graph. Furthermore, none of these works address the effect of substituting movie ratings with tag ratings, leaving the informational value of tag-ratings - whether explicit or inferred - unclear. Conversely, we aim at determining the informational value of explicit descriptive entity ratings, and we conduct experiments to address this issue directly. Furthermore, although tags are plentiful, there is only one type of relationship between a tag and an item, and tags are not inter-related as entities may be in a KG. Finally, none of these datasets are publicly available. *MindReader alleviates these issues by collecting ratings on entities* - recommendable as well as descriptive - drawn from an existing KG where the inter-relations between entities and the semantics of those relations are explicitly defined. Moreover, by representing entities in a KG we can further exploit (direct and indirect) relationships connecting descriptive and recommendable entities, which is not possible with tags.

Ratings for item features. Recent work on preference elicitation [25] highlight the importance of eliciting users' preferences on *rich properties* of items rather than only the items themselves, proposing a "Wizard-of-Oz"-like methodology for effective preference elicitation through dialogue. The user is asked first *what sort of movies* they like, then for an *example* of such movies, and finally *what in particular* was appealing about the provided example movie. Such rich properties of items can be considered analogous to descriptive entities. Their finding supports the potential value of explicit observations from users on descriptive attributes in personalised recommendation tasks and therefore motivates our study of explicit rating on descriptive entities. A similar approach compares metadata-based methods to rating-based methods [24]. Contrasting to the study on preference elicitation [25], this work finds that even

Dataset	#Users	#Entities	#Relation types	#Ratings	Domain	Feedback	Explicit ratings			Available
							Recommendable	Descriptive	Explicit unknowns	
Gedikli et. al [11]	19	100	✗	5,648	Movies	5-star	✔	✔	✗	✗
MovieLens-100K [12]	610	9,724	✗	100,836	Movies	5-star	✔	✗	✗	✔
Tagommender [27]	995	9,724	✗	118,017	Movies	5-star	✔	✔	✗	✗
CCPE-M [25]	502	4,259	✔ (3)	6,297	Movies	Conversational	✔	✗	✗	✔
Netflix [3]	6,769	7,026	✗	116,537	Movies	5-star	✔	✗	✗	✔
MindReader	**1,174**	**10,030**	✔ **(8)**	**102,160**	**Movies**	**Like/Dislike/Unk.**	✔	✔	✔	✔
Das et. al (Cars) [8]	2,180	606	✗	2,180	Cars	Binary	✗	✗	✗	✗
Das et. al (Cameras) [8]	5,647	654	✗	11,468	Cameras	Binary	✗	✗	✗	✗
LibraryThing (DBPedia) [23]	6,789	9,926	✔ (11)	410,199	Books	10-star	✔	✗	✗	✗
BookCrossing [33]	278,858	271,379	✗	1,149,780	Books	10-star	✔	✗	✗	✔
Last.fm [18]	359,347	186,642	✗	17,559,530	Music	Play count	✗	✗	✗	✔

Table 1: Characteristics of MindReader and existing datasets.

a few ratings provide better predictive power than purely metadata-based methods, surprisingly indicating that descriptive feature are less informative than the implicit evidence inferred from movie ratings. Yet, the movie metadata adopted in this work is based purely on word co-occurrence in the movie descriptions. Moreover, user preferences towards these attributes are synthetically inferred from movie ratings as no explicit feedback on words is provided. This differs from our setting where we collect explicit feedback on both items and descriptive entities connected to them (directly and indirectly). Finally, a different approach investigates how *non-content attribute preferences* affect hybrid RSs [20]. In this case, non-content attributes, such as popularity, recency, and similarity to a user previous interactions are leveraged to explain the reason behind some user item selection. Since non-content attributes are derived from item metadata, user sensitivity towards a specific attribute is inferred and not explicitly provided. Interestingly, CF methods are not able to derive preferences from non-content attributes, while hybrid models can exploit these preferences for increased performance. In our work, we also consider how hybrid models can make use of both the entity ratings and the KG entities in generating useful recommendations.

Other data collection platforms. Our data collection application is inspired by *Tinderbook* [23], which provides book recommendations based on few binary ratings provided by the user. Similar to MindReader, Tinderbook (i) exploits an extension of state-of-the-art KG embedding methods, and (ii) relies on an existing knowledge base (DBpedia) to obtain book information. Nonetheless, while *Tinderbook* constitutes a proof-of-concept of how KGs and machine learning can be used for recommendations, no dataset is provided to validate such approaches. MindReader instead provides a reference dataset to support the study of new methods in product recommendation based on ratings of both recommendable and descriptive entities. A widely used dataset is MovieLens [12], which is based on a RS created by the GroupLens Research group. Like MindReader, MovieLens is a RS platform that is also exploited for data collection and has collected a dataset of 25M movie ratings since 1995. Yet, MovieLens contains ratings only on recommendable entities (movies). Moreover, while variants of the MovieLens datasets contains tags similar to descriptive entities, these are free-text, not rated by users, and missing semantic relationships.

A comparison between existing datasets and MindReader can be found in Table 1. In summary, prior works have focused primarily on explicit ratings on user-provided tags, which differs from the descriptive entities we consider in this work as these are provided by an existing and structured knowledge graph. Moreover, existing works try to derive ratings to descriptive entity ratings from recommendable entities. Yet, none of the existing datasets provide explicit ratings on non-recommendable items. A gap that we overcome with the MindReader dataset. Different from existing works, we build the data collection platform MindReader and collect a dataset explicit descriptive entity ratings and study their integration for recommending recommendable entities. Moreover, we not only support explicit *like* and *dislike* ratings, but we also collect explicit *unknown* ratings for those cases in which users were not able to provide a rating (*explicit unknowns*, e.g., a movie not seen, an unknown actor, a category of unclear semantics). Our dataset is freely available and contains more than 100K ratings from 1,174 users over 10K entities connected with rich semantic relationships.

4 METHODOLOGY

To collect the *MindReader dataset*, we developed and published a web-based application called MindReader (https://mindreader.tech/) wherein we follow a gamification approach to elicit user preferences through crowdsourcing over the entities of a movie knowledge graph called the MindReader KG. In MindReader, users are asked about their preferences on movies and possibly related descriptive entities such as actors and genres. At the end, the system tries to guess some user preferences so that users are provided with a list of recommendations as a "reward" for playing the game. Here, we describe how we constructed the MindReader KG (Subsection 4.1) and we outline the overall flow of the data collection application (Subsection 4.2). This application allowed us to collect more than 100, 000 explicit ratings, which we analyse later (Section 5).

4.1 Knowledge graph construction

KGs model how different entities, from one or more domains, are connected, i.e., it models heterogeneous entities and their semantic relationships (as in Figure 1) [21]. Within the movie domain, entities could be movies, genres, and actors. A KG is then a labelled directed multigraph represented as the triple $\langle \mathcal{E}, \mathcal{R}, \mathcal{L} \rangle$. Hence, in a KG the entities are represented with nodes $\mathcal{E}$, names for entities and relationships with labels $\mathcal{L}$, and the relationships are from the set of edges $\mathcal{R} \subseteq \mathcal{E} \times \mathcal{L}' \times \mathcal{E}$, where $\mathcal{L}' \subseteq \mathcal{L}$ captures the relationships types, such that there exists a mapping $\Phi : \mathcal{E} \cup \mathcal{R} \mapsto \mathcal{L}$. Moreover, we say that the set of entities $\mathcal{E}$ is composed by two sets: one of *recommendable entities* $\mathcal{E}_{rec}$ (e.g., movies) and the other of *descriptive entities* $\mathcal{E}_{desc}$ (e.g., actors, or genres), such that $\mathcal{E} = \mathcal{E}_{rec} \cup \mathcal{E}_{desc}$.

We constructed the MindReader KG over a subset of 9, 000 movies from the MovieLens-100K [12] dataset (as of April 2020). Although larger versions of the dataset are available (e.g., MovieLens-25M [12]

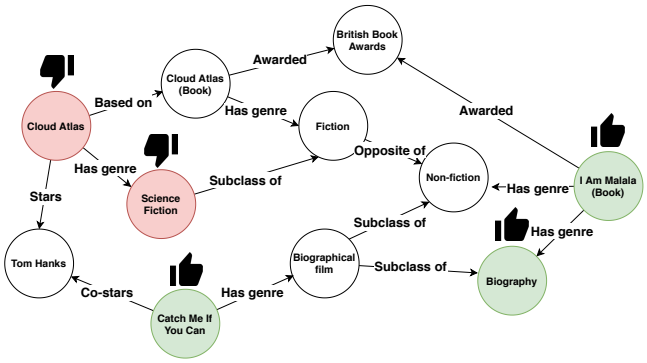

Figure 1: Sample graph with ratings from an hypothetical user.

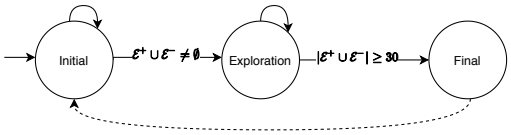

Figure 2: Flow of the MindReader application. The dashed line indicates that users are able to restart the interview upon completion.

with 62k movies), we started with a smaller version, which is still among the most used in the literature [17, 32] to ensure higher coverage of entities (and we plan to employ larger versions in our next version of the dataset). Moreover, even though the MovieLens dataset contains explicit user ratings, which are useful for traditional recommenders and popularity-based sampling, it is otherwise limited in terms of additional metadata to use as descriptive entities. Specifically, it contains only the release year for each movie as well as a number of user-generated tags.

To extend the KG with descriptive entities, we linked the movies in the MovieLens dataset with those in Wikidata [29] by means of existing shared identifiers (e.g., IMDb IDs, https://www.imdb.com/). Then, for every movie, we obtained from Wikidata its associated actors, directors, production studios, release decade, genres, and main subjects. As the hierarchy of genres and subjects is included in Wikidata, we include that information in the KG as well. Finally, to maintain a higher graph density, we deleted all entities with only one relationship. A sample of the KG which exemplifies the inclusion of hierarchical information can be seen in Figure 1. The final KG is stored in the Neo4j (https://neo4j.com) graph database. We summarise the graph statistics in Table 2.

4.2 The phases of MindReader

To obtain ratings for various types of entities, we design the MindReader application as a game where the user is asked to express an explicit preference toward various entities in the underlying KG with the final goal of receiving a list of movie recommendations. Entities are presented in batches, so that the user is asked to rate an entire batch before moving to a new batch. Moreover, we want to obtain at least some ratings about entities (both recommendable and descriptive) that are connected one to the other. Therefore, the game contains three phases (see Figure 2):

(1) An initial phase where the user is asked exclusively about recommendable entities.
(2) An exploration phase where the user is iteratively asked about entities (both recommendable and descriptive) potentially related to those they have stated their preferences for.
(3) A final recommendation phase where system tries to guess the user's preferences.

In transitioning between these phases, the application maintains a state for each session and each user. In all phases, whenever an entity is presented to the user, the user can provide three explicit feedbacks: *like*, *dislike*, or *unknown*. In the state representation, the system keeps track the entities the user likes ($\mathcal{E}^+$), dislikes ($\mathcal{E}^-$), as well as those entities the user states they do not know how to rate ($\mathcal{E}^?$), with $\mathcal{E}^\Omega = \mathcal{E}^+ \cup \mathcal{E}^- \cup \mathcal{E}^?$ being all entities rated by the user. Also, we ensure the user is asked to rate an entity at most once.

During phases 1 and 2, the user is shown one or more batches of 9 entities. The small number has been selected to keep the list of entities to rate short and the feedback process enjoyable for the user. After each batch of entities is rated completely, the session and user states are updated. If the phase termination criteria is met (see Figure 2), the application moves to the next phase, otherwise the same phase continues with a new batch.

Initial phase. The purpose of the initial phase is to gather feedback on a seed set of recommendable entities. The termination criteria for the initial phase is $\mathcal{E}^+ \cup \mathcal{E}^- \neq \emptyset$, after which the interview proceeds to the exploration phase (phase 2). When sampling the entities in the initial phase we attempted to optimise the probability that the user knows at least one of the entities in the list. As a proxy for popularity, we used the number of ratings provided in the MovieLens dataset. The rating distribution of MovieLens is skewed towards movies from the mid-1990s, which some preliminary analysis showed to be caused by a large number of MovieLens users not being active in the platform in recent years. While older movies with many ratings are indeed popular, users usually have a fresher memory regarding more recent movies. As such, we conduct movie sampling while both considering popularity and recency. Thus, for a movie $i \in \mathcal{E}_{rec}$, we define a weight function $W(i) = |Ratings(i)| * \max(1, Age(i) - 2000)$ where $Ratings(i)$ are the ratings for i in the MovieLens dataset, and $Age(i)$ is the number of years since the release of i. When sampling a movie i for user u, this is sampled with probability proportional to $W(i)$ normalised over the sum of weights of all the movies not already rated by the user. While different sampling criteria could be adopted, our initial analysis with few test users suggested this formula to provide a reasonable balance between popularity and recency as to reduce the number of unknown movies.

Exploration phase. In the exploration phase, we collect explicit ratings both for descriptive and recommendable entities until the user has rated at least 30 entities of any type, i.e., $|\mathcal{E}^+ \cup \mathcal{E}^-| \geq 30$. In this phase, the user is asked about entities adjacent (i.e. at 1-hop distance in either direction in the KG) to those in $\mathcal{E}^-$ and $\mathcal{E}^+$. In order to diversify the entities asked about, each round has three sets of three entities sampled independently among those adjacent to $\mathcal{E}^-$, to $\mathcal{E}^+$, and to a set of random recommendable entities sampled again based on the weighting of W, respectively. Among all the entities adjacent to the given seed set, we conduct a weighted sampling based on their global PageRank [22] in order to elicit ratings for more popular entities. To balance the coverage of entities

of all types, we split the adjacent entities into their respective types (e.g., genre, decade, actor) and randomly sample from each of these splits.

Recommendation phase. In the recommendation phase we generate two lists of recommendable entities for the user based on their previous answers: one with entities that we guess they like, and one with entities that we guess they dislike. We then ask the user for feedback on all these entities. The main purpose of this phase is not to provide recommendations of high quality, but rather to collect more ratings. Nevertheless, by providing recommendations to the user as a form of reward, we hope to motivate continued use of the application. To determine which entities to recommend, we follow a sampling approach similar to the one used in the exploration phase. Given the entities $\mathcal{E}^+$, we consider their adjacent recommendable entities and take the top-25 entities by number of connections to entities in $\mathcal{E}^+$. This number has been selected in order to reduce computational load. From this subset, we sample recommendable entities by weight-based sampling on global PageRank as described previously. We repeat the same process for entities in $\mathcal{E}^-$. Moreover, to increase coverage and avoid recommendations only related to the initial seed, an additional set of pseudo-randomly sampled entities is presented to the user. When all rating are collected, we thank the user for their participation and invite them to take the interview again. All subsequent interview answers will be associated with the same user based on a token stored in their browser's local storage. This allows us to avoid storing any user personal information except for their preferences.

5 ANALYSIS

In this section we analyse the data collected from the MindReader application. We present both the characteristics of the graph and the ratings we obtained. *We highlight that this is the first version of the dataset* and our plan is *to proceed with a continuous data collection so to release larger datasets over time.* At the time of writing, the estimated growth is of ~80K new ratings every 3 months. We consider only ratings from users who completed the quiz till the final phase. We consider two variants of our dataset: MR-ALL which contains all observed ratings, and MR-BIN which is a subset of MR-ALL in which we only consider the binary like/dislike ratings.

Knowledge graph statistics. The graph contains more than 18K nodes and 198K edges (Table 2). It includes edges from movies (the recommendable entities) to its adjacent descriptive entities as well as among descriptive entities in a hierarchy to their superclasses. All nodes are directly related to at least 4 other entities in the KG, while the median degree is 10. Moreover, we see that there exists a number of entities that are highly connected since the maximum degree is more 4.4K and the average degree is 21. Finally, there is only one connected component in the KG. That is, there is an undirected path from any entity to any other entity in the KG.

Long tail rating distribution. The MindReader dataset presents a classical long-tail distribution of ratings (Figure 3), which is similar to that of the original MovieLens dataset. Therefore, a small fraction of entities are popular – and receive most positive ratings (the short head) – while the remaining majority received fewer ratings and fewer preferences as well (the long tail). In Figure 3, the entities on the vertical axis are sorted by popularity with the most popular entities at the bottom, i.e., at 1% is the top-1% most popular entities by number of ratings. Therefore, for the MR-BIN dataset, we observe that 20% of ratings involve 1.98% of the most popular entities (≈ 57 entities). Note that long-tail distributions are usually common, and for this reason trivial approaches to recommend the top-n entities in the short head are usually a competitive baseline [7].

Co-rated entities. In order to test collaborative filtering (CF) approaches, we would require that users with similar rating patterns like similar entities [20]. For testing such methods then, it is necessary to evaluate entity co-ratings. Therefore, we consider the number of entities rated by the same user-pair, i.e., for a user-pair $\{u, v\} \in \{\{u, v\} | u, v \in \mathcal{U} \text{ and } u \neq v\}$ we consider the number of entities that both user u and v have rated. Having many user-pairs with a high number of co-rated entities theoretically increases the performance of CF methods. Also, we provide this analysis only for MR-BIN, since "don't know" ratings would not be commonly used in CF approaches (see Figure 4 and Figure 5). While users of MovieLens have a minimum of 25 ratings and a mean of 165 ratings per user, the MindReader game stops in few steps with a mean of 31 ratings per user (87 if we consider "don't know" ratings). For this reason co-ratings for recommendable entities (REs) are generally under 20, while co-ratings for descriptive entities (DEs) are a bit higher (mostly under 50). The reason for higher co-ratings of descriptive entities is twofold. First, there are DEs that are very central in our KG, e.g., the *Drama* and *Action* genres, and those are shown to the users more frequently. Moreover, as we will see later, users are more likely to provide "don't know" rating for REs than DEs. This is particularly important when, as we show later, we are able to infer user preference from DEs instead of REs. This allows us to infer the same amount of information by asking users to rate a smaller amount of entities that are descriptive entities and for which we are more likely to obtain relevant feedback.

Coverage. Figure 6 shows the coverage in terms of rating for different entity types. As many person entities are both actors and directors, we have listed these as "Person". Similarly, many subjects are also genres, hence we refer to the union of these as "Category". We include the fraction of entities for which there are no observations, the fraction of entities for which there are only "Don't know" observations, and the fraction of those for which there are binary ratings (i.e., at least one user liked or disliked the entity). For entities with both a "Don't know" observation and a binary rating, we include it only in the fraction of entities with binary ratings. As expected from the sampling approach we employ, coverage is generally higher for entity types with fewer entities. The exception is movies which has almost full coverage when considering "Don't know" observations. This is because movies can be sampled in all phases, while DEs are shown only in the second phase.

Ratings distribution. We extract separately the recommendable and descriptive entity ratings and consider the distribution of ratings among types of entities. The results of this analysis is shown in Figure 7. On average, 61.8% of movie ratings in an arbitrary MindReader session are "Don't know" ratings. Conversely, on average, only 21.3% of genre ratings in an arbitrary MindReader session are "Don't know" ratings. Naturally, even though users are not familiar with a large number of movies, they still largely have opinions on genres. We see a similar trend for broader qualities of movies, e.g.,

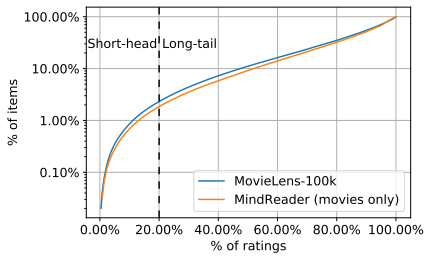

Figure 3: Rating distributions for the MR-BIN and MovieLens datasets.

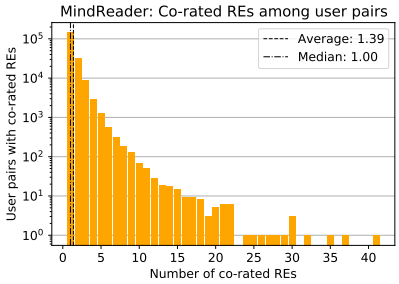

Figure 4: MR user pairs that have co-rated a number of Recommendable Entitiess (REs).

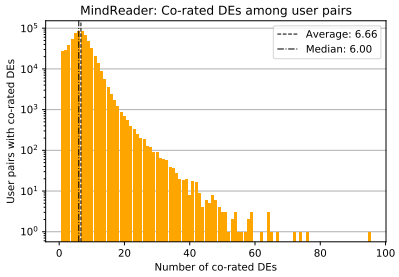

Figure 5: MR user pairs that have co-rated a number of Descriptive Entitiess (DEs).

subjects and categories, while questions towards actors, directors and film companies lead to useful feedback only in smaller proportion (this relates also to the current coverage in the dataset, as seen earlier in Figure 6). This is particularly important for cold-start interviewing systems [6, 25] that conduct interviews with new users to establish their preferential profiles. These interviews must both be short to keep the experience enjoyable for the user, and rich in information as to produce high-quality recommendations based on the inferred preferences of the user. As such, in constructing the interview, the system should minimise the chance that a user is not aware of or has no opinion on a given entity they are asked about, while also ensuring that answers to those questions yield as much useful information as possible.

6 EVALUATING THE IMPACT OF DESCRIPTIVE ENTITIES

We now present a preliminary evaluation of the effects of including explicit user feedback for descriptive entities in a recommendation model. We evaluate a wide range of models comprising naive [7], centrality-based [22, 28], neighbourhood-based [7], and various embedding-based [4, 10, 31] models for recommendation. Specifically, we are interested in observing the effect with respect to the quality of recommendations for recommendable entities (movies) only when extending the model to include information about non-recommendable entities.

6.1 Evaluation setup

We consider ratings as a categorical value $\in$ {Like, Dislike} and use data from the MR-BIN dataset. We leave the evaluation of the potential value of explicit "Don't know" ratings as future work. Moreover, since this first edition of the MindReader dataset is still quite sparse in ratings, we follow the evaluation strategy of other recommender systems in presence of sparse ratings and evaluate all models using Leave-One-Out (LOO) evaluation [7, 9, 14]. To evaluate the quality of the recommendations made, we use k-bounded ranking-based performance metrics [13, 14]. For every user, we generate a ranked list of the k recommendable entities we expect the user to prefer the most. As metric we employ Hit Ratio (HR) and Discounted Cumulative Gain (DCG) at k [13, 14] and we compute them as follows:

$$\text{HR@}k = \frac{\textit{Number of hits @ } k}{\textit{Number of users}} \tag{1}$$

$$\text{DCG@}k = \sum_{i=1}^{k} \frac{2^{rel(i)} - 1}{\log_2(i+1)} \tag{2}$$

where $rel(i) = 1$ when the entity at rank i is the entity the user liked and $rel(i) = 0$ otherwise. Therefore, higher HR@k and DCG@k scores correlate with a higher quality of recommendations.

6.2 Experiments

Given the typical long-tail distribution in most recommendation contexts, most recent studies have focused in designing methods able to provide non-trivial recommendations [7]. Thus, similar to the literature, we perform both tests with the full dataset as well as where we remove the top 2% of popular entities from the test set. This allows studying the degree to which descriptive entity ratings help in generating recommendations for non-trivial entities. Experiment results where the most popular entities are included are described in the appendix. Moreover, we compare the models over 3 different setups, namely:

(a) Adding all ratings. In the first setup we study the difference in performance of the models first only on recommendable entity ratings and then when training on all entity (both recommendable and descriptive) ratings.

(b) Substituting recommendable entity ratings. We test how well descriptive entity ratings can replace those of recommendable entities. Thus, we train the models on datasets with a varying ratio between recommendable and descriptive entity ratings. Specifically, let N_u be the number of recommendable entity ratings made by a user u. Let $M_u(n), D_u(n)$ be functions that sample, at random, n recommendable and descriptive entity ratings, respectively, from the observed ratings of u. Given integers $m = 4$ and $n \in \{3, 2, 1\}$, we train the models on datasets containing the ratings

$$\bigcup_u M_u(\frac{n}{m} \cdot N_u) \cup D_u(\frac{m-n}{m} \cdot N_u)$$

(c) Removing recommendable entities ratings. Finally, we conduct a final experiment where, using the same splits as in experiment **(b)**, we simply remove movie ratings from the training set without substituting them with descriptive entity ratings.

Reproducibility. To support the reproducibility of our results, we publish all our experimental data, setup, and code for running all experiments. Due to space constraints, here we report only results where we remove top popular entities from the test set and only for

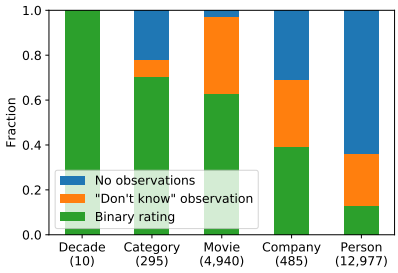

Figure 6: Coverage of different entity types from the KG. The number of entities of the different types is shown in parentheses.

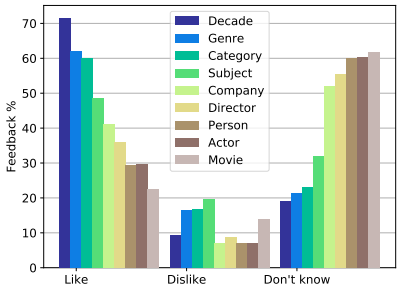

Figure 7: Rating distribution over the entity categories. The feedback percentage is the mean distribution over all user sessions.

Measure	Value
# nodes	18,707
# decades	10
# companies	485
# categories	295
# movies	4,940
# people	12,977
# edges	198,452
Minimum degree	4
Median degree	10
Average degree	21
Maximum degree	4,454
# connected components	1

Table 2: Statistics on the constructed KG.

the first two setups. All other results are described in the extended version of this work on https://mindreader.tech/dataset/.

6.3 Recommendation models

We tested the classical baseline, **TopPop**, a naive approach which recommends the recommendable entity with the most ratings. Additional, we included two Matrix Factorisation (MF) [10] methods. The first, **MF**, a standard model trained using Alternating Least Squares (ALS) updates, and the second **BPR**, trained by maximising the Bayesian Personalised Ranking (BPR) objective [26]. Moreover, we also tested two methods based on kNN. **User kNN**, a kNN-based model recommending entities to users based on preferences of k most similar users, and **Item kNN**, recommending entities to a user based on the k entities most similar to those rated by the user.

In addition, we tested 4 different variations of graph embeddings. **TransE** [4], recommending entities to users based on predicted probability of the user and entity being linked in a rating graph, trained on rating triples only (edges connecting users to rated entities). **TransE-KG**, based on TransE, trained on both rating triples and other triples from the MindReader KG, performing link-prediction on the MindReader KG extended with user-rating-entity triples. Similar to TransE, we also tested two models based on a TransH [31], the base **TransH** model trained on rating triples only and **TransH-KG** including also triples from the MindReader KG.

Finally, we tested 3 recommendation techniques based on the Personalised PageRank (PPR) score [16]. The first method, **PPR-KG**, recommending movies through PPR with the highest PPR score navigating only the MindReader KG, using entities that a user has liked as seeds. The second method, **PPR-COLLAB**, navigating a *collaborative graph* with nodes users and entities and edges connecting only users and their rated entities, that is without any KG edge. The third method, **PPR-JOINT**, finally navigates both the MindReader KGs and the co-rating edges.

6.4 Results

Adding descriptive entity ratings. The results of experiment **(a)** are shown in the "All movies/entities" columns of Table 3. Models were trained only on movie ratings (the "All movies" column), and movie- as well as descriptive entity ratings (the "All entities" column). Statistical significance is determined from a paired sample t-test between the two settings.

First considering model performance when provided with descriptive entity ratings in addition to movie ratings, *we observe that most learning models achieve a statistically significant improvement in HR@k and DCG@k when provided with descriptive entity ratings.* While this improvement in performance may not be of much surprise as we are essentially increasing the amount of training data for each model, it still indicates that the model is able to infer useful information also from explicit ratings on descriptive entities. Moreover, we observe that some models such as Item k Nearest Neighbour (kNN) are especially good at making use of the added ratings, seeing 105.8% improvement in HR@k. Interestingly, we note that while the translational TransH models see a statistical significant increase in performance, the TransE models decrease, possibly due to the fact the that the TransH models are more expressive and better capture relations than TransE models.

Considering the HR and DCG on movie ratings alone in Table 3, we observe that TopPop achieves the highest HR. Other works in the literature have also shown that simply recommending the most popular recommendable entities to users is a surprisingly effective strategy [7]. Nevertheless, while TopPop performs well, when we include descriptive entities the ranking quality by the KG-based PPR models, among others, becomes far superior to this baseline. In fact, PPR-JOINT and PPR-KG are the best performing models when provided with descriptive entity ratings in HR and DCG respectively when compared to all other methods.

Substituting with descriptive entity ratings. The results of experiment **(b)** are shown in the "n/4" columns of Table 3. In each table, all listed models were trained on datasets with shifting ratios between the number of movie and descriptive entity ratings, where the "3/4" column represents 3/4'ths movie- and 1/4'th descriptive entity ratings. Statistical significance is derived from a paired sample t-test between $n/4$ and "All movies". For all models, when we see a statistical significant improvement in the DCG when adding descriptive entities, we also see the same statistical significant change when substituting ratings. *This indicates that descriptive entities carry more information than recommendable entities.* In particular, Item kNN witnesses a ~47.05% increase in HR, though this is primarily attributed the fact that the density of descriptive entity ratings is higher than that of recommendable entities. In this setup, the best performing models are PPR-COLLAB, PPR-JOINT, and MF. *These results demonstrate the potential of descriptive entity ratings. Since we are now replacing data instead of simply adding more training data, we show there is no informational loss due to this operation.* This would allow interviewing systems to focus on querying the user for opinions on more descriptive entities where

	HR@10					DCG@10				
Models	All movies	All entities	3/4	2/4	1/4	All movies	All entities	3/4	2/4	1/4
BPR	0.36 ± 0.06	0.41 ± 0.02*	0.39 ± 0.04	0.39 ± 0.01	0.37 ± 0.02	0.18 ± 0.02	0.19 ± 0.01	0.19 ± 0.02	0.19 ± 0.01	0.19 ± 0.02
Item kNN	0.17 ± 0.01	0.35 ± 0.01*	0.17 ± 0.01	0.20 ± 0.01*	0.25 ± 0.01*	0.09 ± 0.01	0.19 ± 0.01*	0.09 ± 0.00	0.11 ± 0.01*	0.15 ± 0.01*
MF	0.42 ± 0.01	0.45 ± 0.02*	**0.43 ± 0.02**	**0.42 ± 0.01**	0.41 ± 0.02	**0.20 ± 0.01**	0.22 ± 0.01*	**0.21 ± 0.01**	0.20 ± 0.01	0.20 ± 0.01
PPR-COLLAB	0.42 ± 0.01	0.43 ± 0.01	0.41 ± 0.02	0.41 ± 0.01	**0.44 ± 0.01**	**0.20 ± 0.01**	0.21 ± 0.00	0.19 ± 0.01*	0.20 ± 0.01	0.21 ± 0.01*
PPR-JOINT	0.38 ± 0.01	**0.50 ± 0.01***	0.36 ± 0.02	0.37 ± 0.01	0.40 ± 0.02*	0.19 ± 0.01	0.30 ± 0.01*	**0.21 ± 0.01***	**0.23 ± 0.01***	**0.26 ± 0.01***
PPR-KG	0.25 ± 0.01	0.46 ± 0.01*	0.29 ± 0.02*	0.32 ± 0.01*	0.38 ± 0.02*	0.15 ± 0.01	**0.32 ± 0.01***	0.19 ± 0.01*	0.21 ± 0.01*	0.25 ± 0.01*
TopPop	**0.43 ± 0.01**	0.42 ± 0.01	0.42 ± 0.02	0.40 ± 0.01*	0.40 ± 0.02*	**0.20 ± 0.01**	0.20 ± 0.00	0.19 ± 0.01	0.19 ± 0.01*	0.19 ± 0.01*
TransE	0.33 ± 0.03	0.31 ± 0.04	0.34 ± 0.01	0.32 ± 0.02	0.32 ± 0.01	0.18 ± 0.01	0.17 ± 0.02	0.17 ± 0.01	0.16 ± 0.01	0.17 ± 0.01
TransE-KG	0.28 ± 0.01	0.18 ± 0.01*	0.36 ± 0.02*	0.35 ± 0.01*	0.33 ± 0.02*	0.15 ± 0.01	0.10 ± 0.01*	0.18 ± 0.01*	0.18 ± 0.01*	0.17 ± 0.02*
TransH	0.28 ± 0.04	0.32 ± 0.02*	0.26 ± 0.02	0.31 ± 0.04	0.26 ± 0.03	0.15 ± 0.02	0.17 ± 0.01*	0.14 ± 0.01	0.16 ± 0.03	0.14 ± 0.01*
TransH-KG	0.30 ± 0.04	0.35 ± 0.02*	0.31 ± 0.03	0.29 ± 0.03	0.28 ± 0.03	0.16 ± 0.02	0.18 ± 0.01*	0.17 ± 0.02	0.15 ± 0.01	0.15 ± 0.01
User kNN	0.31 ± 0.02	0.40 ± 0.01*	0.30 ± 0.01	0.32 ± 0.02	0.32 ± 0.02	0.17 ± 0.01	0.21 ± 0.01*	0.17 ± 0.01	0.17 ± 0.01	0.18 ± 0.01

Table 3: {HR, DCG}@10 performance. Statistically significant differences in mean performance between related experiment pairs are marked with a star (*).

the chance of getting a useful answer is higher. Yet, looking at the model performance in absolute values, we highlight how there is ample margin of improvement in the design of models that can exploit the informational value of non-recommendable entities in a knowledge graph.

7 CONCLUSION

We introduce MindReader, the first dataset with explicit user ratings on both recommendable and descriptive entities in a knowledge graph within the domain of movie recommendation. We release both the dataset and the open-source platform for expanding the data-collection in other domains. This will allow further research in KG-enhanced recommendation models. Our analysis of the data shows that users are on average more likely to provide informative ratings towards descriptive entities compared to recommendable entities. This observation is particularly important for interview-based cold-start recommender systems. We also evaluate a variety of models for recommendation to assert the informational value of explicit descriptive entity ratings for recommendation. We find that including descriptive entity ratings in the training process increases the performance of almost all models thus justifying the potential of exploiting this kind of information. In future work we plan to integrate datasets from multiple domains (e.g., books and music) as well as expand the MindReader survey application to collect more organic feedback, e.g., paired preference between entities or free text comments.

Acknowledgements. This work is supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement no. 838216 and the Poul Due Jensen Foundation. We want to thank the MindReader users for their help in building the dataset.

REFERENCES

[1] S. E. Aldrich. 2011. Recommender Systems in Commercial Use. *AI Magazine* (2011), 28.
[2] V. Bellini, A. Schiavone, T. Di Noia, A. Ragone, and E. Di Sciascio. 2018. Knowledge-aware Autoencoders for Explainable Recommender Sytems. (2018), 24–31.
[3] J. Bennett, S. Lanning, et al. 2007. The netflix prize. In *KDD'07*. 35.
[4] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *NIPS'13*. 2787–2795.
[5] Y. Cao, X. Wang, X. He, Z. Hu, and T. Chua. 2019. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In *WWW'19*. 151–161.
[6] K. Christakopoulou, F. Radlinski, and K. Hofmann. 2016. Towards conversational recommender systems. In *KDD'16*. 815–824.
[7] P. Cremonesi, Y. Koren, and R. Turrin. 2010. Performance of recommender algorithms on top-n recommendation tasks. In *RecSys'10*. 39–46.
[8] M. Das, G. De Francisci Morales, A. Gionis, and I. Weber. 2013. Learning to question: leveraging user preferences for shopping advice. In *KDD'13*. 203–211.
[9] M. Deshpande and G. Karypis. 2004. Item-based top-n recommendation algorithms. *TOIS'04* (2004), 143–177.
[10] S. Funk. 2006. Netflix Update: Try This at Home. https://sifter.org/~simon/journal/20061211.html.
[11] F. Gedikli and D. Jannach. 2013. Improving recommendation accuracy based on item-specific tag preferences. *TOIS'13* (2013), 1–19.
[12] F . M. Harper and J. A. Konstan. 2015. The movielens datasets: History and context. *TIIS'15* (2015), 1–19.
[13] X. He, T. Chen, M. Kan, and X. Chen. 2015. TriRank: Review-aware Explainable Recommendation by Modeling Aspects. In *CIKM '15*. 1661–1670.
[14] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. Chua. 2017. Neural collaborative filtering. In *WWW'17*. 173–182.
[15] Y. Hu, Y. Koren, and C. Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *ICDM'08*. 263–272.
[16] G. Jeh and J. Widom. 2003. Scaling personalized web search. In *WWW'03*. 271–279.
[17] Y. Jhamb, T. Ebesu, and Y. Fang. 2018. Attentive contextual denoising autoencoder for recommendation. In *ICTIR'18*. 27–34.
[18] M. Levy and K. Bosteels. 2010. *Music recommendation and the long tail*.
[19] D. Liang, J. Altosaar, L. Charlin, and D. M. Blei. 2016. Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence. In *RecSys'16*. 59–66.
[20] F. Mourão, L. Rocha, J. A. Konstan, and W. Meira. 2013. Exploiting Non-Content Preference Attributes through Hybrid Recommendation Method. In *RecSys'13*. 177–184.
[21] N. Noy, Y. Gao, A. Jain, A. Narayanan, A. Patterson, and J. Taylor. 2019. Industry-Scale Knowledge Graphs: Lessons and Challenges. *ACM Queue* 17, 2 (2019).
[22] L. Page, S. Brin, R. Motwani, and T. Winograd. 1999. *The PageRank citation ranking: Bringing order to the web*. Technical Report. Stanford InfoLab.
[23] E. Palumbo, A. Buzio, A. Gaiardo, G. Rizzo, R. Troncy, and E. Baralis. 2019. Tinderbook: Fall in Love with Culture. In *ESWC'19*. 590–605.
[24] I. Pilászy and D. Tikk. 2009. Recommending New Movies: Even a Few Ratings Are More Valuable Than Metadata. In *RecSys'09*. 93–100.
[25] F. Radlinski, K. Balog, B. Byrne, and K. Krishnamoorthi. 2019. Coached Conversational Preference Elicitation: A Case Study in Understanding Movie Preferences. In *SIGDIAL'19*. 353-–360.
[26] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *AUAI'09*. 452–461.
[27] S. Sen, J. Vig, and J. Riedl. 2009. Tagommenders: Connecting users to items through tags. In *WWW'09*. 671–680.
[28] B. Shams and S. Haratizadeh. 2017. Graph-based collaborative ranking. *ESWA'17* (2017), 59–70.
[29] D. Vrandečić and M. Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85.
[30] H. Wang, N. Wang, and D. Yeung. 2015. Collaborative Deep Learning for Recommender Systems. In *KDD'15*. 1235–1244.
[31] Z. Wang, J. Zhang, J. Feng, and Z. Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *AAAI'14*. 1112–1119.
[32] S. Zhang, L. Yao, and X. Xu. 2017. AutoSVD++ An Efficient Hybrid Collaborative Filtering Model via Contractive Auto-encoders. In *SIGIR'18*. 957–960.
[33] C. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. 2005. Improving recommendation lists through topic diversification. In *WWW'05*. 22–32.