Mining Latent Features of Knowledge Graphs for Predicting Missing Relations

Tobias Weller, Tobias Dillig, Maribel Acosta, and York Sure-Vetter

Karlsruhe Institute of Technology
Englerstraße 11, Karlsruhe, Germany
{tobias.weller, maribel.acosta, york-sure-vetter}@kit.edu

Abstract. Knowledge Graphs (KGs) model statements as head-relation-tail triples. Intrinsically, KGs are assumed incomplete especially when knowledge is represented under the Open World Assumption. The problem of KG completeness aims at identifying missing values. While some approaches focus on predicting relations between pairs of known nodes in a graph, other solutions have studied the problem of predicting missing entity properties or relations even in the presence of unknown tails. In this work, we address the latter research problem: for a given head entity in a KG, obtain the set of relations which are missing for the entity. To tackle this problem, we present an approach that mines latent information about head entities and their relations in KGs. Our solution combines in a novel way, state-of-the-art techniques from association rule learning and community detection to discover latent groups of relations in KGs. These latent groups are used for predicting missing relations of head entities in a KG. Our results on ten KGs show that our approach is complementary state-of-the-art solutions.

1 Introduction

Knowledge graphs (KGs) have become an important foundation to represent knowledge exploited in, e.g, Question Answering, Entity Linking, and recommender systems. While current real-world KGs, such as DBpedia [3] and Wikidata [20], contain millions of facts, they still suffer from incompleteness which may hinder the effectiveness of the applications where they are consumed.

Motivating Example. Consider the KG depicted on Figure 1 (left), representing facts about persons. However, in the KG, the entity *Paul_Sereno* is not described with the relation *placeOfBirth*. The question that arises is whether the relation *placeOfBirth* for this head entity is missing. Furthermore, it could be that the actual value (i.e., the tail entity) for *placeOfBirth* does not exist in the KG. This work aims at predicting missing relations for a given head entity, regardless of the existence of the tail entity in the KG.

To address the problem of KG completeness, approaches typically assume that two of the components in the triple are known a priori, e.g., when predicting relations, it is assumed that the head and tail entities are represented in the KG. This is a typical assumption in approaches that rely on subsymbolic representations, e.g., KG embeddings [5, 19]. However, predicting missing relations to estimate head entity completeness is not directly possible under this assumption, as shown in the following example. Following

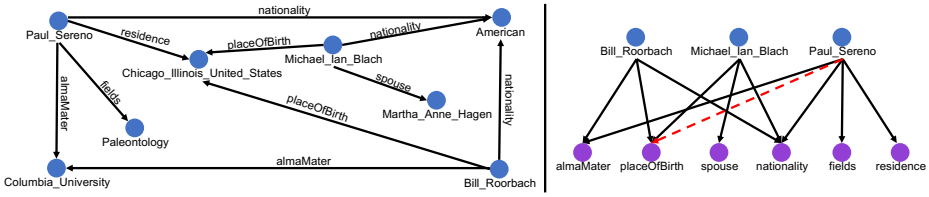

Fig. 1: Motivating Example. Consider the given subgraph of DBpedia on the left. We address the problem of predicting missing relations, based on a given head-entity. On the right we converted the graph into a bipartite graph by only considering the heads and the relations of the KG. The red dotted edge corresponds to the motivated scenario of predicting the missing relation *placeOfBirth* for entity *Paul_Sereno*.

the motivating scenario from Figure 1, assume that the entity *Paul_Sereno* was actually born in *Aurora (Illinois, USA)*, leading to the fact that current approaches cannot detect that the entity *Paul_Sereno* is incomplete with respect to the relation *placeOfBirth*, as they cannot establish an association between a head entity and an unknown tail entity. To overcome this limitation, we propose a solution to perform the prediction of missing relations associated with an entity in a KG even in the presence of unknown tail entities. Our solution comprises two main stages: the relation-centric stage and the prediction stage. In the relation-centric stage, we use a technique from association rule learning to reduce noise and mining groups of frequently occurring relations. Afterwards, the information from that analysis is represented as a graph. This graph is used to identify communities of relations, which represent clusters of latently related relations. In the prediction stage, the information from the interactions of the entities in the KG and the information about the latently related relations encoded in the communities are used to predict missing relations of head entities. Experimental results show that the evaluated approaches exhibit a complementary performance, and that our approach significantly outperforms the state-of-the-art in four out of the ten studied KGs.

2 Related Work

Knowledge Graph Completion. TransE [5] represents relation, head and tail entities as vectors. ComplEx [19] is based on the same concept as TransE, but uses complex-valued vectors for predicting relations in knowledge graphs. Both methods are transductive learning algorithms, making it possible to predict missing parts of triples, given that the individual entities and relations are known to the model in advance. In contrast to these methods, EDMAR [18] and RDF Shape Induction [10] are inductive methods that learn a general model from examples and are therefore applicable to all triples, even if entities and relations were not known in advance while learning the model. In the context of KG completion, there are approaches that rely on the symbolic representation of KGs. HARE [1] is an engine that detects missing values in a KG based on the Local-Closed World Assumption. It crowdsources the missing values to complete the KG and allows for answering SPARQL queries. Other work specifies the number

of missing relations and thus measures the completeness of the KG [14]. The information about missing relations can be used to learn rules [8] for KG completion. All these methods predict relations between two given entity nodes in the KG.

Frequent Itemsets. High-utility Itemsets [7, 9] is an extension to Highly-correlated itemset mining [2] in which the most frequent itemsets are to be found, which yield the highest profit. The utility of the transactions is the most important criteria. However, the problem of finding high-utility frequent itemsets is computational very expensive. Faster High-Utility Itemset Mining (FHM) is a very fast High-Utility Itemset Mining algorithm [7], which reduces the number of join operations and thus improves the runtime. However, utilizing utility results in many itemsets which yield a high utility but correlate only very weakly. Therefore, FHM was extended to guarantee that the itemsets correlate strongly, besides yielding a high utility [7].

Community Detection. The identification of communities is particularly prominent in the area of social network analysis [17]. Community detection, however, is not exclusively applicable to social networks. Network analyses in the field of co-authorship are also conceivable [12]. In general, a community is a dense subgraph. Detecting them is computationally very expensive. For this reason, random walks [15] or grouping methods [16] have been used to simplify and speed up the computations. However, a trade-off will arise here between the quality of the results and runtime for large networks. Nevertheless, there are also approaches that have returned reliable results on very large graphs while exhibiting a satisfying runtime [6].

3 Problem Definition

In this work, we define a knowledge graph G as $G = (H \cup T, R)$, where H denotes the set of head entities, T the set of tail entities, and R the set of labelled relations. The information in the knowledge graph G can be modeled as triples (h, r, t), with $h \in H$ denotes the head entity which has a relation $r \in R$ to a tail entity, denoted as $t \in T$. Furthermore, consider $R_h(G)$ the set of relations where the entity h appears in the head of a statement in G, i.e., $R_h(G) = \{r \mid \exists t \in T, (h, r, t) \in G, r \in R\}$.

Problem Statement. Given a knowledge graph G, consider G^* the ideal graph, containing all statements known about entities that should be in G, i.e., $G \subseteq G^*$. For a given head entity $h \in H$ in G, the research problem is to identify the set of missing relations of h, i.e., the set of relations defined as $R_h(G^*) \setminus R_h(G)$.

4 Our Approach

An overview of our proposed solution to predict missing relations is presented in Figure 2. In our proposed solution, we distinguish two main stages: the *relation-centric stage* and the *prediction stage*. The *relation-centric stage* captures the latent features of the relations encoded in the KG. The outcome of this stage is then used in the *prediction stage* to predict missing relations of head entities.

Fig. 2: Proposed approach for the predictions of missing relations for head entities, based on a knowledge graph G. The relation-centric stage captures latent knowledge between the relations. The prediction stage predicts missing relations based on the communities detected in the previous stage and the KG G for a given head entity h.

4.1 Relation-Centric Stage: Mining Latent Interactions from Relations

This stage identifies groups of relations that are related based on the implicit knowledge encoded in G. The input of this stage is a KG G. To get a better view on the co-occurrence of relations we transform G into a bipartite graph. In this bipartite graph, nodes represent head entities and relations, while edges encode the head-relation interactions. We denote this graph the *head-relation graph*.

Definition 1 (Head-Relation Graph). *Assume a KG $G = (H \cup T, R)$. A head-relation graph is a bipartite graph $I = (V, E)$, where $V = H \cup R$. An edge $(h, r) \in E$ denotes that the head entity $h \in H$ interacts with the relation $r \in R$ in G.*

To illustrate the concept of a head-relation graph I, consider the running example from Figure 1 (right) of the graph. The head-relation graph corresponds to persons and their existing outgoing relations in the knowledge graph G. Based on I, we will identify highly-correlated relations. To this end, we propose the application of frequent itemsets mining. Frequent itemsets approaches rely on transactions to identify items that highly co-occur. In our approach, the set of all relations of one head entity represents one transaction. Therefore, all transactions can be determined by the union over the transaction of the individual head entities. A good side effect of using frequent itemsets, is the removal of noise in the data and filter out relations that occur very rarely. One important aspect to consider when applying frequent pattern mining is that many frequent patterns are not interesting and items cannot appear more than once in a transaction. This is for the usage of itemset mining in KGs not useful, since a head entity can have the same relation multiple times to different tail entities, e.g. *fields*, and this might affect the computation of itemsets. At the same time some relations which occur very infrequent but are of high interest could be higher weighted than others that occur very frequently in a KG but are at the same time only of limited interest. Using the Apriori algorithm [2] would identify frequent itemsets, but could not overcome those limitations. To overcome those two limitations, the Fast Correlated high-utility itemset Miner (FCHM) [7] efficiently finds highly correlated itemsets, based on transaction data. FCHM prunes all itemsets that does not fulfill a minimum number of utility (*minutil*) and correlation (*minbond*). The bond measure indicates how items in a frequent itemset correlate and thus expresses the relative importance of a relationset [7, 13]. This method allows for identifying relations that correlate and, therefore, occur very frequently with each other. In addition, we can identify and remove relations that occur only very rarely in the KG.

These low occurring relations are noise in the KG and due to their low occurrence provide only very little information for the prediction of relations. The input for FCHM is a set of transactions. One transaction is the list of existing relations for a head entity h, i.e., R_h. Considering our motivating example in Figure 1, the transaction for head entity *Paul_Sereno* is the following (*residence*, *fields*, *nationality*, *almaMater*). The computation of highly-correlated relationsets, using a head-relation graph is defined as follows:

Definition 2 (Highly-Correlated Relationsets). *Let $I = (H \cup R, E)$ be a head-relation graph, $minbond \in \{x \in \mathbb{R} \mid 0 \leq x \leq 1\}$ be a minimum bond threshold and $minutility \in \mathbb{R}^+$ be a minimum utility threshold. The set of interactions is $D = \{T_1, T_2, \ldots, T_q\}$ where each element is a tuple $T_x := \big(X = \{r \in R \mid \exists h_x \in H, (h_x, r) \in E\}, |X|\big)$ containing the relations of head entity $h_x \in H$ and the number of the relations as utility. FCHM receives $minbond$, $minutility$ and D as input parameters and returns the set $\mathcal{S}$. The set $\mathcal{S} = \{S_0, S_1, \ldots, S_m\}$ is a set of highly-correlated relationsets where $S_k \subseteq R$ and $B(S_k) \geq minbond$, for each $S_k \in \mathcal{S}$. $B(S_k)$ denotes the bond measure of the highly-correlated relationset S_k and is defined as follows:*

$$B(S_k) = \frac{support(S_k)}{dissup(S_k)}, \textit{where}$$

$$support(S_k) = |\{h \in H \mid \forall r \in S_k : (h, r) \in E\}|,$$
$$dissup(S_k) = \sum\nolimits_{r \in S_k} |\{h \in H \mid (h, r) \in E\}|.$$

The outcome of FCHM are sets of highly correlated relations, called relationsets. The returned relationsets in $\mathcal{S}$ are different in size and strongly overlapping. Thus, a relation $r \in R$ can occur in different relationsets. Due to the overlap and the differences in the sizes of these sets, the information from the relationsets will be grouped. To extract information from the relationsets, we will model the corresponding relations as nodes in an undirected, weighted graph, which we denote relation-bonding graph G'.

Definition 3 (Relation-Bonding Graph). *Let $\mathcal{S} = \{S_0, S_1, \ldots, S_m\}$ be a set of highly-correlated relationsets. An Relation-Bonding Graph is an undirected weighted graph $G' = (R', E', w)$, where $R' = \bigcup_{S_k \in \mathcal{S}} S_k$, and for each $S_k \in \mathcal{S}$, $r_x, r_y \in S_k \Rightarrow (r_x, r_y) \in E'$. The weights w are defined as a function $w : R' \times R' \to \mathbb{R}$ and computed as the sum of the corresponding bond measure, i.e.:*

$$w(r_x, r_y) = \sum_{r_x, r_y \in S_k} B(S_k).$$

The graph G' contains the relations from the relationsets as nodes. It should be noted that due to the computation of the highly-correlated relationsets, $R' \subseteq R$ applies, which means that not every item $r \in R$ of the original graph G must also be represented in G'. The edges of G' represent the common occurrence of relations in the same relationset. The weight of the edge between the relations is the sum of all bond values of the relationsets in which both relations occur. The weight of the edge thus expresses the strength of its tie across all relationsets. In the last step to determine the latent features from the relations, we will use the information represented in the relation-bonding

graph G' to identify communities within. A community is a set of nodes in a graph such that each node of the set is densely connected to each other node in the set. The identification of communities in G' is used to group relations that are strongly related.

There are many community detection algorithms that use different methods, e.g., minimum cut method or modularity maximization. In particular, the modularity describes the strength of a network by dividing it into communities. We chose Fastgreedy algorithm [6] for detecting communities, which optimizes the metric modularity when discovering communities. A benefit of Fastgreedy is that there is no need to predefine the number of communities since this algorithm detects the best number of communities by itself. Fastgreedy is a non-overlapping community detection algorithm, which means that nodes in the graph are exactly assigned to one community. By using the Fastgreedy algorithm, communities will be detected in the relation-bonding graph G'. These communities represent a set of relations from G' that have a high density, with only a few connections to the other communities. The communities represent latent features mined from the KG and, in our work, are called relation communities.

Definition 4 (Relation Community Set). *Let G' be a relation-bonding graph. A relation community set is denoted $\mathcal{C} = \{C_1, C_2, \ldots, C_p\}$, where $C_j \in \mathcal{C}$ is a relation community defined as a dense sub-graph of G', and $C_i \cap C_j = \emptyset$, for each $C_i, C_j \in \mathcal{C}$.*

4.2 Prediction Stage: Predicting Missing Relations in Knowledge Graphs

We use the information from the KG G and the information we mined from it and encoded in the relation community sets $\mathcal{C}$, to predict missing relations of head entities. In general, the number of possible relation candidates for predicting is, depending on the number of relations in the KG, usually very high. Therefore, in the following, we reduce the number of possible candidate relations. For this, we use the information from the community sets $\mathcal{C}$. We compute for a head entity h the relative number of its existing relations in the KG G to each community set. We sort the results in descending order. Exemplary sorted communities for a given head-entity is e.g. $C_1 = \frac{7}{10}, C_2 = \frac{3}{23}, C_3 = \frac{1}{10}, C_4 = 0$. We select the first community set C_1 with the highest relative frequency, unless the relative frequency is one. A relative frequency of one for a community means that the head entity h is already complete with respect to the relations from this community. Possible candidates for missing relations of a head entity h are now all relations in this community set that the entity h does not already have. In mathematical terms, this means that, starting from a fixed h and C_i, we check the following relations as possible candidates for prediction: $R_{cand} = \{r \mid r \in R : r \in C_i \wedge \neg\exists t \in T : (h, r, t) \in G\}$. For each of these candidates we compute a confidence of prediction. The confidence for predicting a relation $r \in R_{cand}$ for head entity $h \in H$ is computed as follows:

$$conf(h, r) = \frac{|\{h_j \mid h_j \in E \wedge \exists t \in T : (h_j, r, t) \in G \wedge \exists r_k \in R, r_k \neq r \exists s, t \in T : (h_j, r_k, s) \wedge (h, r_k, t)\}|}{|\{h_j \mid h_j \in H \wedge \exists t \in T : (h_j, r, t) \in G\}|}$$

The confidence divides the number of head entities that have relation r and share at least one relation with h by the number of entities that have relation r. We compute for each relation in R_{cand} its confidence and use the top-k relations as predictions.

Table 1: Overview of experimental configurations per knowledge graph. At the top of the table, a summary of characteristics and at bottom of the table, parameters used for the computation of communities for our approach.

Metric	FB15k	WN18	Pers(DBp)	Pers(WD)	Comp(DBp)	Comp(WD)	Mov(DBp)	Mov(WD)	Songs(DBp)	Songs(WD)
#Entities	14,951	40,943	229,613	190,419	63,545	10,925	231,637	287,775	39,619	126,606
#Relations	1,345	18	2,239	1,509	1,189	304	959	382	332	321
#Train	483,142	141,442	313,296	229,059	142,887	12,103	396,834	390,295	95,833	184,542
#Valid	50,000	5,000	10,000	10,000	10,000	10,000	10,000	10,000	10,000	10,000
#Test	59,071	5,000	10,000	10,000	10,000	10,000	10,000	10,000	10,000	10,000
minbond	0.1	0	0.1	0.1	0.1	0.1	0.1	0.3	0.1	0.1
minutility	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

5 Experiments

Datasets. We used DBpedia (DBp) [3] and Wikidata (WD) [20] for the evaluation. We used subgraphs related to the class Person (Pers), Company (Comp), Movie (Mov) and Song. In addition, we used FB15k [4] and WN18 [11] for evaluating our approach. An overview of the characteristics of the KGs used in this evaluation is given in Table 1.

Silver Standards. We constructed silver standards for each of the previously described KGs. We split the KGs into three disjunctive sets: training, validation, and test set. We call it silver standard, as the created test sets may suffer from incompleteness originated in the KG, thus, creating spurious false positives. In other words, a prediction may be correct but the relations might be missing in the KG and hence in the test set.

Configurations. We set the utility for computing frequent relationsets to a constant value of 1 ,i.e, each relation is considered equally important. The $minbond$ where chosen such that the relations of the union of all highly-correlated relationsets covers 70%–90% of the relations in G. In this way, we make sure that the information loss is minimized and at the same time enable a sufficient removal of noise in the data. This resulted in $minbond$ values from 0 to 0.3 for the different KGs. The used parameters of the entire experimental setup are given in Table 1.

Metrics. Following related KG completion studies, we use Hits@k as evaluation metric. Hits@k measures the proportion of correct relations in top-k ranked relations.

Preprocessing. For DBpedia and Wikidata, we removed regularly appearing relations for all head entities, e.g., *wikiPageID*, *wikiPageRevisionID*, and *P31*.[1] We removed them for making predictions more challenging by deleting regularly occurring relations.

5.1 Comparison to Related Knowledge Graph Completion Approaches

We selected TransE [5] and ComplEx [19] for comparison, since they gained a lot of momentum in the area of KG completion and achieve very good results on prominent knowledge graph completion tasks. We used the default parameters for TransE and ComplEx for all KGs. It is import to note that, unlike our method, both methods use

[1] These are DBpedia- and Wikipedia-specific relations to denote information about the Wikipedia page and the class of an entity, respectively.

Table 2: Comparison of our approach with state-of-the-art algorithms. Our approach (CPP) uses the head entity to predict missing relations. The compared methods uses head and tail entity to predict missing relations.

	Hits@1			Hits@3			Hits@10		
KG	CPP	TransE	ComplEx	CPP	TransE	ComplEx	CPP	TransE	ComplEx
FB15k	.389	.667	**.519**	.473	**.885**	.800	.698	**.974**	.940
WN18	.561	.924	**.945**	.650	.974	**.986**	.900	**.997**	.995
Pers(DBp)	**.438**	.085	.085	**.490**	.149	.233	**.655**	.246	.292
Pers(WD)	.253	**.328**	.273	.254	.431	**.468**	.367	.517	**.618**
Comp(DBp)	.275	.185	**.319**	.345	.326	**.699**	.580	.452	**.780**
Comp(WD)	**.635**	.483	.008	**.647**	.603	.017	.674	**.692**	.058
Mov(DBp)	.393	**.453**	.106	**.615**	.515	.222	**.900**	.582	.347
Mov(WD)	**.471**	.383	.205	**.567**	.450	.424	**.833**	.527	.553
Songs(DBp)	.398	**.444**	.409	.498	**.898**	.736	.811	**.980**	.887
Songs(WD)	.488	**.788**	.203	.654	**.941**	.359	.825	**.986**	.452

head and tail information to predict relations. The additional information about the tail entity for predicting the missing relation is not available to our approach. This must be taken into account when analysing the results, which are presented in Table 2. The results show that even without the information about the tail entity, our approach is competitive with state-of-the-art methods. For some KGs, our approach achieved higher values in the Hits@k metric than the compared methods. Our solution is superior to the other methods in KGs with a high number of relations, as is the case of Pers(DBp). At the same time, in Pers(DBp), the mean size of communities and the standard deviation is very low. This ensures that there are fewer relations in the individual communities and thus the predictions become more precise. Considering the Pers(WD) KG, the number of relations is also very high, but the structure of the computed communities is not as compact as the community structure of Pers(DBp) KG. The mean size of communities in Pers(WD) is higher (cf. Table 3), as well as the standard deviation. As a result, the Hits@k performance is lower, compared to the other methods. Another consideration in this evaluation is that not all the relations in the silver standard are covered by the communities computed by our approach. As can be observed in Table 3, except for the WN18, not all the relations are present in the detected communities. This coverage varies from 37.07% to 86.47%, which hinders the number of true positives achieved by our approach. Therefore, in the following experiment, we analyze the performance of our solution when the silver standard only contains relations from the communities.

5.2 Comparison with Relations in Community

To demonstrate the relevance of the latent structures of the relations in the communities, the following are examples of relation communities for FB15k detected with our proposed approach: *Community 18* = {*nominated_for, honored_for, award_nominee}*, and *Community 25* = {*symptom_of, diseases, causes, risk_factors}*.

Table 3: Overview of the structure of computed communities for the studied KGs.

Metric	FB15k	WN18	Pers(DBp)	Pers(WD)	Comp(DBp)	Comp(WD)	Mov(DBp)	Mov(WD)	Songs(DBp)	Songs(WD)
Relations	86.47%	100%	72.53%	84.84%	78.47%	71.38%	51.02%	37.07%	77.27%	44.59%
Communities	110	3	244	76	77	28	104	31	29	31
mean Com.	10.57	6	6.52	16.58	12.12	7.75	4.55	4.48	8.21	4.52
std. Com.	15.49	4	16.9	44.83	39.53	10.30	4.32	8.60	10.28	4.68
max Com.	71	10	213	313	313	47	40	44	39	24
min Com.	2	2	2	2	2	2	2	2	2	2

In this evaluation, we study the results of our approach if we only consider relations in the test set which are present in the communities. To this end, we will filter out triples that contain relations which are not present in the computed communities for the respective KGs. We report again on the Hits@k metric (cf. Table 4), since it allows for a better comparison with the results from the previous section. As expected, the overall performance of our approach increases, since now only known relations are now considered. For Hits@1, our approach now significantly improves for FB15k. With increasing k, however, the gain in Hits@k turns out not to be strong in the same proportion. The result of WN18 does not change at all, since all relations are covered in the communities. For the KGs Pers(DBp) and Pers(WD), the performance of our approach does not significantly increase. We hypothesize that the structure of the communities, more precisely the very large average size and standard deviation, is the reason for this. Similar observations apply to Comp(DBp). The largest improvement in Hits@k can be observed in Comp(WD), which has a moderate coverage of relations (71.38%, cf. Table 3). The standard deviation of the communities is slightly increased, but the dataset has a low average community size. The results of this KG increase significantly due to the reduction to known relations. The results for both Mov(DBp) and Mov(WD) do not increase significantly, although the average size of the communities is very small, and the coverage of the relations is moderate to low, respectively. However, our approach outperforms the state-of-the-art in these KGs in the previous evaluation (Hits@3, Hits@10), which indicates that the original silver standard already includes a high number of relations that are covered by the communities. Likewise, there is no strong effect on the results of Songs(DBp). However, Songs(WD) benefits from the adjustment of the test dataset. Both, the amount of covered relations in the communities is low, as well as the average number of relations per community. Restricting the data and the analysis to relations known in the communities leads to an improvement of the results.

5.3 Discussion of Experimental Results

While TransE and ComplEx exploit head and tail entity information to predict the missing relation information, our approach uses only the head entity to perform the predictions. Therefore, a direct comparison is difficult. However, in order to position our empirical results with respect to state-of-the-art solutions, we still compared against KG embedding methods despite the differences in the underlying assumptions. The experimental results show that the evaluated approaches exhibit complementary Hits@k performance, i.e., there is no single approach that outperforms the others in all the KGs.

Table 4: Performance of the proposed approach over the filtered test dataset. The datasets contain triples which have a relation that is present in a relation community.

KG	FB15k	WN18	Pers(DBp)	Pers(WD)	Comp(DBp)	Comp(WD)	Mov(DBp)	Mov(WD)	Songs(DBp)	Songs(WD)
Hits@1	.394	.561	.462	.254	.277	.733	.404	.486	.400	.502
Hits@3	.483	.650	.520	.257	.350	.775	.634	.599	.503	.675
Hits@10	.714	.900	.695	.371	.589	.803	.927	.886	.821	.852

The advantage of our method is the usage of only head entity information. Therefore, missing relations, even to unknown tail entities, can be predicted. In general, we observe that the performance of our approach strongly depends on the structure of the computed communities and the number of relations in the KG. A small number of average relations per community and a small deviation from the average allows for achieving better results with our approach. This effect can be followed by looking at the average community size (cf. Table 3) and the results of the Hits@k (cf. Table 2).

Another important consideration is the original incompleteness of the KGs. Consider the movie *The_Naked_Gun* in the Mov(DBp) dataset. We predicted among others *basedOn* as missing relation for this head entity. According to our silver standard, this prediction is considered to be a false positive because this head entity does not contain a *basedOn* relation in the KG. However, assuming complete knowledge, the prediction of our approach would be correct, because the film is based on the American television comedy *Police Squad!*. Similar cases of spurious false positives are encountered in other KGs, e.g. Pers(DBp). For example, the head entity *Deven_Marrero* describes an American professional baseball player. Our approach predicted *throws* as missing relation, which was wrongly considered a false positive. The above examples illustrate the problems involved in evaluating KG completeness. Although in some cases the predictions are correct, the evaluation classifies them as wrong since the information is not available in the KG. Due to the incompleteness of the KGs, the actual quality of the predictions cannot be assessed with absolute certainty.

6 Conclusions and Future Work

In this paper, we presented an approach to predict missing relations for head entities in Knowledge Graphs (KG). Our approach groups related relations by means of latent relationships encoded by the interactions of the head entities with their corresponding relations. These associations are exploited for detecting communities of frequent co-occurring relations in the KG. The experimental results show that our approach is competitive with existing KG embedding approaches (TransE and ComplEx), even if they use information about the tail entity for the prediction. We observed that our approach can keep-up and compete (for the metric Hits@k) with existing KG embedding methods that uses head and tail entities for predicting missing relations.

Future work could focus on further structures of communities, as these have a high impact on the performance of our solution. Furthermore, our approach could be integrated into a larger KG completion pipeline that is able to: (i) predict missing relations

for head entities with our approach; (ii) identify tail candidates (for known entities), using for example TransE or ComplEx, based on the predictions from the previous step.

References

1. Acosta, M., Simperl, E., Flöck, F., Vidal, M.: Enhancing answer completeness of SPARQL queries via crowdsourcing. J. Web Semant. **45**, 41–62 (2017)
2. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proceedings of VLDB '94. pp. 487–499. Morgan Kaufmann Publishers Inc. (1994)
3. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: A Nucleus for a Web of Open Data. In: Proceedings of ISWC. pp. 722–735. Springer (2007)
4. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: A collaboratively created graph database for structuring human knowledge. In: Proceedings of ACM SIGMOD. pp. 1247–1250. ACM (2008)
5. Bordes, A., Usunier, N., Garcia-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Proceedings of NIPS. pp. 2787–2795 (2013)
6. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. Phys. Rev. E **70** (Dec 2004)
7. Fournier-Viger, P., Wu, C.W., Zida, S., Tseng, V.S.: Fhm: Faster high-utility itemset mining using estimated utility co-occurrence pruning. In: Foundations of Intelligent Systems. pp. 83–92. Springer (2014)
8. Galárraga, L.A., Teflioudi, C., Hose, K., Suchanek, F.: Amie: Association rule mining under incomplete evidence in ontological knowledge bases. In: Proceedings of WWW. pp. 413–422. ACM (2013)
9. Liu, Y., Liao, W.k., Choudhary, A.: A fast high utility itemsets mining algorithm. In: Proceedings of UBDM '05. pp. 90–99. ACM (2005)
10. Mihindukulasooriya, N., Rashid, M.R.A., Rizzo, G., García-Castro, R., Corcho, O., Torchiano, M.: Rdf shape induction using knowledge base profiling. In: Proceedings of ACM SAC. p. 1952–1959. SAC '18, Association for Computing Machinery, New York, NY, USA (2018)
11. Miller, G.A.: Wordnet: A lexical database for english. Commun. ACM **38**(11), 39–41 (Nov 1995)
12. Newman, M.E.J.: The structure of scientific collaboration networks. Proceedings of the National Academy of Sciences **98**(2), 404–409 (2001)
13. Omiecinski, E.R.: Alternative interest measures for mining associations in databases. IEEE TKDE **15**(1), 57–69 (Jan 2003)
14. Pellissier Tanon, T., Stepanova, D., Razniewski, S., Mirza, P., Weikum, G.: Completeness-aware rule learning from knowledge graphs. In: Proceedings of ISWC. pp. 507–525 (2017)
15. Pons, P., Latapy, M.: Computing communities in large networks using random walks. In: Proceedings of ISCIS'05 (2005)
16. Raghavan, U.N., Albert, R., Kumara, S.: Near linear time algorithm to detect community structures in large-scale networks. Phys. Rev. E **76** (Sep 2007)
17. Rebhi, W., Yahia, N.B., Ben Saoud, N.B.: Hybrid modeling approach for contextualized community detection in multilayer social network. Procedia Comput. Sci. **112**(C), 673–682 (Sep 2017)
18. Tran, M.D., d'Amato, C., Nguyen, B.T., Tettamanzi, A.G.B.: An evolutionary algorithm for discovering multi-relational association rules in the semantic web. In: Proceedings of GECCO. p. 513–520. GECCO '17, ACM, New York, NY, USA (2017)
19. Trouillon, T., Welbl, J., Riedel, S., Gaussier, E., Bouchard, G.: Complex embeddings for simple link prediction. In: Proceedings of ICML. pp. 2071–2080. JMLR.org (2016)
20. Vrandečić, D., Krötzsch, M.: Wikidata: A free collaborative knowledgebase. Commun. ACM **57**(10), 78–85 (Sep 2014)