Fine-grained Evaluation of Rule- and Embedding-based Systems for Knowledge Graph Completion

Christian Meilicke, Manuel Fink, Yanjie Wang,
Daniel Ruffinelli, Rainer Gemulla, and Heiner Stuckenschmidt

Research Group Data and Web Science, University of Mannheim, Germany

Abstract. Over the recent years, embedding methods have attracted increasing focus as a means for knowledge graph completion. Similarly, rule-based systems have been studied for this task in the past. What is missing so far is a common evaluation that includes more than one type of method. We close this gap by comparing representatives of both types of systems in a frequently used evaluation protocol. Leveraging the explanatory qualities of rule-based systems, we present a fine-grained evaluation that gives insight into characteristics of the most popular datasets and points out the different strengths and shortcomings of the examined approaches. Our results show that models such as TransE, RESCAL or HolE have problems in solving certain types of completion tasks that can be solved by a rule-based approach with high precision. At the same time, there are other completion tasks that are difficult for rule-based systems. Motivated by these insights, we combine both families of approaches via ensemble learning. The results support our assumption that the two methods complement each other in a beneficial way.

1 Introduction

Knowledge graph completion or link prediction refers to the task of predicting missing information in a knowledge graph. A knowledge graph is a graph where a node represents an entity and an edge is annotated with a label that denotes a relation. A directed edge from s to o labelled with r corresponds to a triple $\langle s, r, o\rangle$. Such a triple can be understood as the fact that subject s is in relation r to object o. As a logical formula we write $r(s, o)$. Often knowledge graphs are created automatically from incomplete data sources that do not fully capture the real relations between the entities. The goal of knowledge graph completion is to use the existing knowledge to find these correct missing links without adding any wrong information. The current evaluation practice estimates model performance by the model's ability to complete incomplete triples like $\langle s, r, ?\rangle$ or $\langle ?, r, o\rangle$ derived from a known fact $\langle s, r, o\rangle$. The task in this case consists of generating a candidate ranking for the empty position that minimizes the amount of wrong suggestions ranked above the correct ones.

Recently, a new family of models for knowledge graph completion has received increasing attention. These models are based on embedding the knowledge graph into a low dimensional space. A prominent example is TransE [2], where both nodes (entities) and edge labels (relations) are mapped to vectors in $\mathbb{R}^n$. Other examples include RESCAL [9], TransH [16], TransG [17], DistMult [18], HolE [8] or ProjE [13]. Once the embeddings have been computed, they can be leveraged to generate a candidate ranking

for the missing entity of a completion task. Over the last years many different models have been proposed that follow this principle.

In contrast, rule-based approaches learn logical formulas that are the explicit representation of statistical regularities and dependencies encoded in the knowledge graph. To predict candidates for incomplete triples, the learned rules are applied to rank candidates based on the confidence of the rules that fired. Works that focus on embeddings usually do not compare the proposed models with rule-based methods and vice versa. In this paper, we do not present a substantially novel method for knowledge graph completion. Instead, we apply AMIE [4], an existing system for learning rules, as well as our own approach called RuleN to this problem. The development of RuleN is mainly inspired by the idea of using a very simple mechanism that can be completely described in the paper. In our experiments, we have found that on the datasets commonly used for the evaluation of embedding based models, both systems are highly competitive. Among the many different embedding-based models for which results have been reported over the recent years (see [6, 12]), only few exceptions performed better.

In a rule-based approach each generated candidate comes with an explanation in terms of the rule that generated this candidate. With the help of these explanations, we analyze the datasets commonly used for the evaluation of embeddings by partitioning their test set. Each subset is associated with the type of the rule which generated the correct test triple with high confidence, e.g., a *symmetry* or *subsumption* rule. This analysis sheds light on the characteristics and difficulty of these datasets. Based on this partitioning, we compare the performance of various rule- and embedding-based approaches (RESCAL [9], TransE [2] and HoleE [8]) on a fine-grained level. Our results show that a large fraction of the test cases is covered by simple rules that have a high confidence. These test cases can be solved easily by a rule-based approach, while the embedding models generate clearly inferior results.

There is also a fraction of test cases that is hard for rule-based approaches. We use the method from [15] to learn an ensemble including both types of approaches. Our results show that the ensemble can achieve better results than the top-performing approach on each dataset used in our experiments. This confirms our findings that both families of approaches are strong on different types of completion tasks, which can be leveraged by the ensemble.

2 Related Work

Within this section, we first discuss methods for learning rules. We continue with approaches that use observed features, which correspond to certain types of rules, to learn a model. Note that there is no clear distinction between the first and the second group of approaches. Finally, we explain latent feature models that are based on the idea of using embeddings and we give some details on the three models we used in our experiments.

Regarding rule-based methods for relational learning, Quickfoil [19] is a highly scalable ILP algorithm that mines first order rules for given target relations. Quickfoil is in principle designed to learn rules that strictly hold. While it also tolerates a small amount of noise, i.e., it can also learn rules even though there are some negative examples in the given knowledge base, it cannot learn rules with a low confidence. However, these

rules are also important for ranking the candidates of a knowledge completion task. In many cases, we may not have a strict rule, but only weak evidence.

AMIE [4] is an approach for learning rules that is similar to our approach introduced in the next section as RuleN. It has a different language bias, as explained in more detail in Section 3.1. The main difference is that AMIE computes the confidence based on the whole knowledge graph, while our approach will compute an approximation that is based on selecting a random sample. It can be expected that AMIE is complete and that the confidences of AMIE are precise. This is not the case for RuleN. However, due to the underlying sampling mechanism RuleN might be able to mine longer path rules. We use AMIE in our experiments as an alternative approach for learning rules.

The path ranking algorithm [7] (PRA) is based on the idea of using random walks to find characteristic paths that frequently occur next to links of a target relation. These paths are used as features in a matrix where each row corresponds to a pair of entities. By including negative examples generated under the Closed World Assumption, a logistic regression is performed on the matrix to train a classifier. The classifier for a relation can then be used to predict the likelihood of the target relation between two given entities based on the surrounding path features. The rule bodies in RuleN correspond to the paths in PRA. While PRA puts a lot of emphasis on learning how to combine the path features with machine learning, RuleN is simpler in this regard. It uses the path features in a more conservative way for which it approximates the significance of invidivual paths more thoroughly. A more expressive extension of PRA is presented in [5], where the authors extract further sub-graph features besides paths.

In [10], Niepert proposes Gaifman Models. Gaifman Models are a way of sampling small subgraphs from a knowledge graph in order to learn a model that uses first order rules as features. One of the main differences is that the set of features, which needs to be defined prior to learning the model, comprises all possible rules of a certain type. Contrary to this, RuleN stores only those rules for which we found at least one positive example during sampling. In the experiments presented in [10] the authors use all path features of length 1 and path features of length 2 that use only one relation in the rule body (e.g., rules that express transitivity of a relation), which corresponds to a subset of the rules that AMIE or RuleN can learn.

Another approach that uses observed features has been proposed in [14]. As feature set the authors use path features of length 1 and features that reflect how probable it is for a certain entity to appear in subject/object position of a certain relation. The latter correspond to the constant rules of RuleN. The authors show that such a model can score surprisingly well on the commonly used datasets, which motivates them to propose the FB15k-237 dataset that we will consider in our experiments. The results are compared against several approaches that are based on embeddings. This analysis (observed vs. latent features) is similar to our evaluation effort. However, we use AMIE and RuleN to learn rules that are more expressive than the feature sets used in [14] and [10] without the need for negative examples. Furthermore, we perform a more fine-grained evaluation based on the distinction between different types of completion tasks.

It has already been argued that a simple rule-based approach restricted to learning inverse relations can achieve state-of-the-art results on WN18 and FB15k [3]. Our evaluation extends these findings by partitioning the “easy” test triples into detailed catego-

ries, which allow fine-grained insight into the performance of different systems. Also, the Inverse Model in [3] is too simple to represent the state-of-the-art performance of rule-based systems on FB15-237.

In contrast to methods which exploit observed features or rules, latent feature models learn representations of the entities and relations from the knowledge base in a low-dimensional space, such that the structure of the knowledge base is represented in this latent space. These learned representations are known as the *embeddings* of the entities and relations, respectively. The models provide a score function $f(s, r, o)$ which for a given triple $\langle s, r, o\rangle$ reflects the model's confidence in the truthfulness of the triple. Based on this, potential candidates for a given query $\langle s, r, ?\rangle$ can be ranked.

Our comparisons in this work focus on bilinear models, which have been successful in the standard benchmarks for this task. RESCAL [9] is a factorization-based bilinear model. It represents entities as vectors $\mathbf{a}_i \in \mathbb{R}^n$, relations as matrices $\mathbf{R}_k \in \mathbb{R}^{n\times n}$ and has a score function $f(s, r, o) = \mathbf{a}_s^T \mathbf{R}_r \mathbf{a}_o$. HolE [8] represents entities as vectors $\mathbf{a}_i \in \mathbb{R}^n$, relations as vectors $\mathbf{r}_k \in \mathbb{R}^n$ and has a score function $f(s, r, o) = \mathbf{r}_r^T(\mathbf{a}_s \star \mathbf{a}_o)$, where $\star$ refers to the circular correlation between $\mathbf{a}_s$ and $\mathbf{a}_o$. TransE is a translation-based model, which represents entities as vectors $\mathbf{a}_i \in \mathbb{R}^n$, relations as vectors $\mathbf{r}_k \in \mathbb{R}^n$ and has a score function $f(s, r, o) = \|\mathbf{a}_s + \mathbf{r}_r - \mathbf{a}_o\|_2^2$.

3 A Simple Rule-based Approach

In this work, we are interested in understanding which types of rules help in knowledge base completion and can be applied successfully to the datasets currently used for evaluating state of the art methods. For this goal, we developed our own rule-based system RuleN that is simple enough to be described in detail within this work. It is based on learning the types of rules defined in Section 3.1 with a sampling strategy described in Section 3.2. In Section 3.3 we explain how to apply the learned rules to rank the candidates for a given completion task.

3.1 Types of Rules

Let r and s refer to relations, x and y to variables that quantify over entities, and let a be a constant that refers to an entity. RuleN supports the following types of rules:

$$r(x_1, x_{n+1}) \leftarrow s_1(x_1, x_2) \wedge \ldots \wedge s_n(x_n, x_{n+1}) \qquad (P_n)$$

$$r(x, a) \leftarrow \exists y\, r(x, y) \qquad (\mathrm{C})$$

We call rules of type P_n with $n \geq 1$ path rules. Given two entities x_1 and x_{n+1} that are connected by an r-edge, a path rule describes an alternative path that leads from x_1 to x_{n+1}. Note that a path in this sense may also contain edges implicitly given by the inverse relations, e.g. $s_3^{-1}(x_3, x_4)$ corresponds to $s_3(x_4, x_3)$. Type C rules are rules with a constant in the head of the rule. The language bias introduced by these rule types is similar to that of existing systems such as PRA [7] and AMIE [4] but there are differences. For example, AMIE does not limit constants to the head of a rule and is in general slightly more expressive. However, it does not learn rules of type C. Concrete

examples for some of these rule types are shown in the following. These rules have been generated in the experiments that we report about later.

$$hyponym(x, y) \leftarrow hypernym(y, x) \quad [0.94] \quad (1)$$

$$celebrityBreakup(x, y) \leftarrow celebrityMarriage(x, y) \quad [0.08] \quad (2)$$

$$producedBy(x, z) \leftarrow sequel(x, y) \wedge producedBy(y, z) \quad [0.55] \quad (3)$$

$$language(x, English) \leftarrow \exists y\ language(x, y) \quad [0.64] \quad (4)$$

Rule 1 and 2 are examples of type P_1. The latter depicts the fact that 8% of celebrity marriages in that dataset ended in divorce. Rule 3 is an example for type P_2. Rule 4 is an example for rule type C that captures that in 64% of the cases, the spoken language of a person is English.

3.2 Learning Rules

For a given rule R, let $h(R) = r(x, y)$ denote its head and $b(R)$ denote its body. As defined in [4], the head coverage is the number of $h(R) \wedge b(R)$ groundings that can be found in the given knowledge graph, divided by the number of $h(R)$ groundings. A head coverage close to 100% suggests that the rule can be used to propose candidates for most completion tasks of relation r. The confidence of a rule is defined as the number of $h(R) \wedge b(R)$ groundings divided by the number of $b(R)$ groundings. Confidence tells us how likely it is that a candidate proposal generated by this rule is correct.

To learn rules for a target relation r, RuleN utilizes a twofold sampling approach instead of a complete search. We first explain the learning of path rules of maximum length n. Given a target relation r, we need to find rule bodies $b(R)$ for $r(x_1, x_{n+1}) \leftarrow b(R)$ that result in helpful rules. The straightforward approach is to look at all triples $\langle a, r, b\rangle$ in the training set and determine all possible paths up to length n between a and b each time using an iterative deepening depth-first search. Using these paths as body for the rule, the confidence can be calculated in a second step. To speed up this rule finding step, it is only performed for k (= sample size) triples $\langle a, r, b\rangle$. Each rule that is constructed this way has a head coverage > 0. Moreover, the higher the head coverage of a rule, the more likely it is to be found. For example, a rule with a head coverage of 0.01 will be found for $k = 100$ with a probability $\approx 63.4\%$. This illustrates that the procedure can miss rules with a low head coverage.

We apply a similar approach for C rules. Given a target relation r, we randomly pick k facts $\langle a, r, b\rangle$. For each of these facts, we create the rules $r(x, b) \leftarrow r(x, y)$ and $r(a, y) \leftarrow r(x, y)$. An example is Rule 4.

In a second step, we compute the confidence of path rules by randomly sampling true body groundings. We then approximate the factual confidence by dividing the number of groundings for which the head is also true by the total number of groundings sampled for the body. With respect to a C rule, we simply pick a sample of r facts and count how often we find a or b in subject and object position.

3.3 Applying Rules

Given a completion task $\langle a, r, ?\rangle$, we select all rules with r in their head. Suppose that we have learned four relevant rules as shown in Table 1. For each of the three path rules,

Table 1. Four relevant rules for the completion task $\langle a, r, ? \rangle$ resulting in the ranking $\langle g(0.81), d(0.81), e(0.23), f(0.23), c(0.15) \rangle$.

Rule	Type	Confidence	Result
$r(x,y) \leftarrow s(y,x)$	P_1	0.81	$\{d, g\}$
$r(x,y) \leftarrow r(y,x)$	P_1	0.70	$\emptyset$
$r(x,y) \leftarrow t(x,z) \wedge u(z,y)$	P_2	0.23	$\{e, f, g\}$
$r(x,c) \leftarrow \exists y\, r(x,y)$	C	0.15	$\{c\}$

we look up all body groundings in the given KB where we replace x by a, collecting all possible values of the variable y. For the constant rule, the body is implicitly true when using the rule to make a prediction for the object position, so it is not checked. What this simply means is that the rule always predicts the constant c when asked for the object position of r, independent of the subject.

A rule can generate one candidate (fourth row), several candidates (first and third row), or no candidate (second row). There are different ways to aggregate the results generated by the rules. As a basis, we choose the most robust approach. We define the final score of an entity as the maximum of the confidence scores of all rules that generated this entity. If a candidate has been generated by more than one rule, we use the amount of these rules as a secondary sorting attribute among candidates with the same (maximum) score. Hence g is ranked before d in the given example. Combining confidences of multiple rules for the same candidate in a more sophisticated way is difficult due to unknown probabilistic dependencies between rules. For example, we found that an aggregation based on multiplication distorts the results (e.g., when two rules of which one subsumes the other fire simultaneously), leading to worse predictions.

4 Experimental Results

Within our experiments we focussed mainly on the three datasets that have been extensively used to evaluate embedding-based models for knowledge graph completion: the WordNet dataset WN18 described in [1], the FB15k dataset, which is a subset of FreeBase, described in [2] and FB15k-237, which has been designed in [14] as a harder and more realistic variant of the FB15k dataset. FB15k-237 is also known as FB15KSelected. We published additional evaluation results for WN18RR, which is a harder variant of WN18 without inverse relations proposed in [3] online at `http://web.informatik.uni-mannheim.de/RuleN/`. The web page contains also the RuleN code and other relevant material.

First, we computed results for the two rule-based systems AMIE and RuleN. Our results imply that rule-based systems are competitive and that it is easy to determine settings for them which yield good results. Next, we divided the datasets into partitions to perform a fine-grained evaluation including TransE, RESCAL and HolE, as well as AMIE and RuleN. Finally, we evaluated an ensemble of these five systems showing that this is a way to leverage the strengths of both approaches.

We followed the evaluation protocol proposed in [2]. Each dataset consists of a training, validation and test set which are used for training, hyperparameter tuning and

evaluation respectively. Each triple $\langle s, r, o \rangle$ from the test set results in two completion tasks $\langle ?, r, o \rangle$ and $\langle s, r, ? \rangle$ that are used to query the systems for a ranked list of entities for the placeholder. hits@k is the fraction of completion tasks for which the removed entity was ranked at least at rank k. We only looked at filtered hits@k, which means that for each completion task, entities other than the removed one which also result in true triples contained in the dataset, are ignored in the ranked list. The filtered mean reciprocal rank MRR is calculated by summing over all completion tasks the reciprocals of the ranks of the removed candidate after filtering.

4.1 Performance of Rule-based Approaches

Embedding-based models have hyperparameters which need to be optimized on a validation dataset. Rule-based systems also have hyperparameters. However, in our experiments, we found them easy to set for knowledge base completion even without a validation dataset. As these hyperparameters are typically a mechanism to tune running time versus completeness of the rule learning process, we simply used the most expressive setting that still finishes within a reasonable time.

The hyperparameters of RuleN are the sample size and the length of the path rules. For AMIE, we focused on thresholds for support, head coverage, and rule length. Furthermore, for both systems, it is possible to disable the mining of rules with constants. In our experiments, we have found that there is indeed a positive correlation between setting the hyperparameters as liberally as possible and the prediction performance. (The only exception to this paradigm resulted in a performance drop of less than 1%.) In Table 2, we show the filtered hits@10 results for increasingly liberal settings for runtimes < 10 h on WN18 and FB15k.

RuleN has one sampling parameter that affects the number of mined rules and one that determines the precision of the confidence calculation. We tied both to the same value, which we varied between 50 and 1000. It is interesting to see that there seems to be a limit for the sample size of RuleN above which the performance remained stable and that it was possible to achieve very good results already with a low sample size and consequently a low run time. Note that this enables RuleN to be applicable to very large (in number of entities) knowledge graphs as long as the number of relations is bounded.

An overview on the results that current state of the art approaches achieve on these two datasets can be found in [12] and [6]. In summary, for WN18 there are only few approaches that achieved a hits@10 score higher than 95%, e.g. 96.4% (Inverse Model [3]), 96.4% (R-GCN+ [11]), 95.5% (ConvE [3]) and 95.3% (IRN [12]). The follow-up approaches scored around 92-95%, while there are still many other approaches that achieved less. For the FB15k dataset there is a higher variance in the results. The best approaches achieved a hits@10 score of 92.7% (IRN [12]) and 88.2% (TransG [17]). However the vast majority could not top a score of 84%. Thus, RuleN and AMIE outperformed the majority of models for which results have been reported on WN18 and FB15k. On FB15k there are only few systems that achieved better results and none of them perform better on WN18. These results show that symbolic representations can compete with and perform sometimes better than many of the approaches that are based on embeddings. This insight is not only supported by the good results of RuleN, but

Table 2. Impact of different settings on performance of rule-based systems. For RuleN, the number in the *Setting* column denotes the sample size. For AMIE, it shows the values for support (s) and head coverage (hc) used for the mining of the path and constant rules respectively. The length of rules with constants was set individually for AMIE as denoted by the *Rule Type* column.

			FB15k				WN18			
	Rule Type	Setting	hits@10	Learn	Apply	#Rules	hits@10	Learn	Apply	#Rules
RuleN	P_{12}	50	.853	1167s	137s	69k	.943	5s	5s	230
	P_{12}	100	.859	2491s	165s	96k	.943	8s	5s	314
	P_{12}	500	.862	6120s	170s	158k	.945	22s	5s	693
	P_{12}	1000	.862	6492s	207s	177k	.945	34s	6s	945
	C	1000	.312	1s	25s	94k	.05	1s	10s	12k
	P_{12}, C	1000	**.875**	6493s	191s	270k	.948	6s	12s	13k
	$P_{12[3]}, C$	1000[100]	.870	49868s	10272s	917k	**.958**	398s	20s	41k
	$P_{123[45]}, C$	1000[100]					**.958**	4103s	151s	54k
AMIE	P_{12}, C_1	s=0, hc=0.0/0.01	.858	4889s	1952s	861k	.942	17s	4s	352
	P_{123}, C_2	s=0, hc=0.0/0.01					.948	868s	29s	4806

especially by the competitive results of AMIE, which we could use almost out of the box to generate the presented results.

All experiments were performed on a machine with 4 cores at 2394 Mhz and 8 GB memory. Even in the most complex setting reported in Table 2, we were able to run the rule-based systems in a few hours on FB15k. Runtimes on FB15k-237 were slightly shorter than those on FB15k as it is a subset of it. For the WN18 dataset, there are competitive settings where we finished in less than a minute, including learning and prediction. It would take much longer on this hardware setting to train the embedding-based models to competitive performance. In our experiments, we found that rule-based systems were orders of magnitude faster to train due to the required hyperparameter search of embedding-based models. The training and prediction runtimes for a given hyperparameter setting were comparable to rule-based systems though.

4.2 Dataset Partitioning

In the following we examine each of the datasets in detail. In particular, we analyze which types of rules are relevant to correctly predict the missing information in the test sets of these datasets. For that purpose, we restricted RuleN to learn P_1 and P_2 rules only. We further distinguish between special sub-types of these rules as follows:

- We refer to a rule of form $r(x, y) \leftarrow r(y, x)$ as a **symmetry rule**. An example is $married(x, y) \leftarrow married(y, x)$.
- We refer to a rule of form $r(x, y) \leftarrow s(x, y)$ with $r \neq s$ as an **equivalence rule** if the reverse direction $s(x, y) \leftarrow r(x, y)$ holds also.[1]

[1] We annotate a rule as equivalence rule if it holds in both directions with a *similar confidence*. We said that two confidence values are similar if they do not differ more than 0.05. This is a pragmatic decision, which allows us to define a meaningful category.

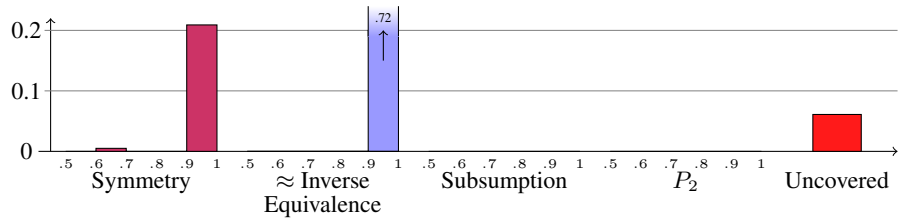

Fig. 1. Rule coverage for the WN18 dataset. We truncated the y-axis; the majority of the test cases are covered by inverse equivalence (72%).

- We distinguish in the case of equivalence between **inverse equivalence**, i.e. $r(x, y) \leftarrow s(y, x)$, and plain **equivalence**. An example for an inverse equivalence rule is $hypernym(x, y) \leftrightarrow hyponym(y, x)$.
- We call any P_1 rule that is not a symmetry or (inverse) equivalence rule a **subsumption rule**, e.g., $cityIn(x, y) \leftarrow capitalOf(x, y)$.

We used RuleN with a sample size of 1000 to learn P_1 and P_2 rules for both WN18 and FB15k. Then we removed all rules with a confidence lower than 0.5. We applied this very restrictive set of rules to the completion tasks defined by the test sets. For each completion task we applied all relevant rules in descending order with respect to their confidence. If one of the candidates generated by the rule was the entity replaced with a question mark, we marked the completion task as solved by the type of that specific rule. Note that we did not continue to check the remaining rules. Thus, we annotated each completion task with the type of the most confident rule that could solve the task. This annotation follows the naming convention defined above. If we could not find such a rule, we annotated the task as *Uncovered*. It is not the case that a completion task annotated as *Uncovered* cannot be correctly predicted by a rule-based approach. There is still the possibility that it can be correctly solved by a rule with low confidence or by a rule which is not of type P_1 or P_2.

In Figure 1, we have depicted the results of applying this approach to the WN18 dataset. The dataset has very specific characteristics. Only $\approx 6.12\%$ of the completion tasks fall into the *Uncovered* category. Moreover, the majority of the tasks is covered by equivalence rules (72.5%). Note that we have grouped the rules of each type with respect to their confidence in the ranges from $(0.5, 0.6]$ to $(0.9, 1.0]$. Here, all of the inverse equivalence rules have a confidence higher than 0.9. An example of an equivalence rule that dominates the dataset is Rule 1 (together with its reversed counterpart) that we already presented above. The remaining tasks are covered by symmetry rules. An example for such a rule is $see\_also(x, y) \leftarrow see\_also(y, x)$. Again, most of them are highly confident. It is also interesting to see that subsumption and P_2 rules do not help to detect anything that is not already covered by equivalence or symmetry rules with higher confidence. For that reason, any method that is capable of exploiting equivalence and symmetry should be able to find the correct candidate for $\approx$94% of the test cases.

The results of applying our approach to the FB15k dataset are shown in Figure 2. For this dataset we observe a heterogeneous set of rules that covers a smaller fraction (still

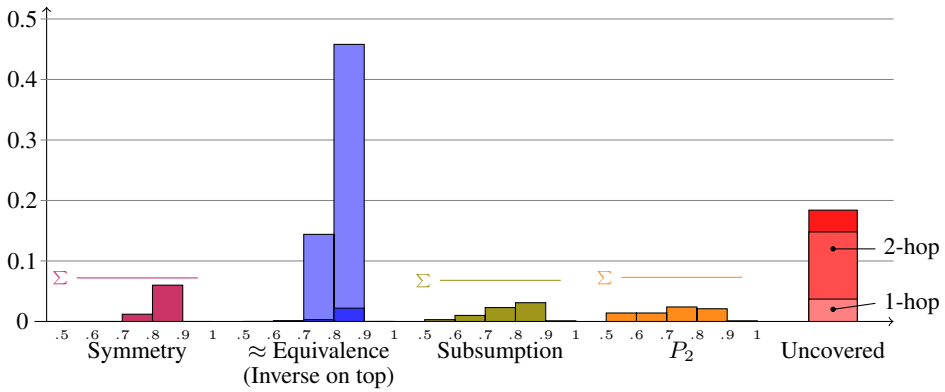

Fig. 2. Rule coverage for the FB15k dataset. Σ shows the total fraction of a specific rule type.

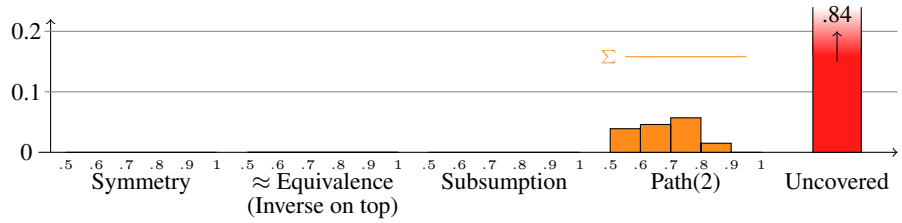

Fig. 3. Rule coverage for the FB15k-237 dataset. 31% of the Uncovered category are ≥2-hop testcases, 69 % are 2-hop testcases, and none of them are 1-hop testcases.

81.6%) of the tasks in the test set. The dataset is still dominated by equivalence (dark blue) and especially inverse equivalence (light blue) rules. These rules cover around 60% of all completion tasks. However, we find now also subsumption rules (6.8%), that are not equivalence or symmetry rules, and P_2 rules (7.3%). Moreover, the fraction of uncovered tasks (18.4%) is larger compared to WN18, but still rather small.

This time, we also analyzed the uncovered tasks in more detail and further divided it into three subgroups. If such a completion task is based on reconstructing a triple $\langle a, r, b\rangle$, we determined the shortest path between a and b in the training set. In Figure 2 we distinguish between 1-hop, 2-hop and other test cases (where the shortest path between a and b has a length ≥ 3). Note that for 1-hop and 2-hop test cases there is still a chance that rules of length 1 or 2 can be used to find the correct candidates. However, since these test cases are not in one of the other categories, we know that those rules would have a confidence lower than 50%.

The high fraction of test cases covered by simple rules could give the impression that WN18 and FB15k are too easy. FB15k-237 has been designed in [14] as a harder variant of the FB15k dataset by making the following two modifications. First, all (inverse) equivalent relations have been removed from the dataset resulting in a knowledge graph with 237 remaining relations. Second, the validation and test sets were changed,

such that any triple $\langle x, r, y\rangle$ is removed from it, if there is some other triple $\langle x, s, y\rangle$ or $\langle y, s, x\rangle$ with $s \neq r$ or $s = r$ in the training set, i.e. x and y are connected by a direct edge in the training set. Figure 3 illustrates the impact of these modifications. The first modification suppresses any kind of dependencies in the dataset that would be captured by (inverse) equivalence rules. The second modification is even more aggressive, because it suppresses any dependencies that could have been exploited by any kind of P_1 rule. These modifications result in a harder dataset, while at the same time introducing an unrealistic bias. Suppose the test set of a dataset with the modifications of FB15k-237 contains a completion task like $murdered(?, john)$. Then it is impossible that the correct murderer of $john$ is his brother, his wife, his boss, his employee, or any person directly related to him in any way. What makes this circumstance really problematic is the fact that the training set may well include examples of murders for which there is another direct relationship between the subject and object. Hence, any system that correctly learns this pattern from the examples in the training set will be penalized for it in the common evaluation format, as including directly related entities in the candidate ranking for a test case can only worsen the performance but never improve it. Therefore, results on FB15k-237 need to be taken with a grain of salt, especially if a system makes any use of P_1 rules. Indeed, we found that suppressing all P_1 rules, the performance of AMIE on FB15k-237 actually improved by roughly 2% for hits@10. For the FB15k-237 results presented in this paper, however, we always used the full rule set.

4.3 Fine-Grained Evaluation

In the following, we present results for each of the annotated subsets. We used AMIE and RuleN with the most liberal settings described in Table 2. As approaches that are based on the use of embeddings, we used the methods TransE [2], RESCAL [9], and HolE [8], for which we did a hyperparameter search as described in [15]. The so-found best hyperparameters are available online. The evaluation results are depicted in Tables 3, 4, and 5. The shortcuts Sym, Eq, Sub, and UC in the table headings refer to the subsets *Symmetry*, *Equivalence*, *Subsumption* and *Uncovered*. We focus mainly on the FB15k dataset because it covers completion tasks from all subsets.

The best performing embeddings based system (HolE) achieved only 36% in terms of hits@1 on FB15k, while AMIE and RuleN achieved 64.7% and 77.2%. The interesting aspect is not the hits@1 itself, but the pattern that if the rule-based systems presented the correct candidate within the top 10, it was usually on the first position. This is not the case for the embedding models. In the *Symmetry* category, for example, the first candidate of TransE was always wrong. We found that for a completion task like $\langle a, r, ?\rangle$, the highest ranked entity was always a itself. This problem with symmetry was less severe for HolE and RESCAL, however, the tendency is the same.

For the subsets *Equivalence*, *Subsumption*, and P_2, RuleN and AMIE could not generate results close to 100% anymore. However, they were still significantly ahead in terms of hits@10 and especially hits@1 score. On WN18, HolE was a noteworthy exception as it achieved competitive results to RuleN and AMIE on the mentioned subsets. TransE and RESCAL performed worse. If we look at the FB15k *Uncovered* subset, we observed a different pattern. Rule-based and embedding-based approaches performed on a similar level with respect to the hits@10 score.

Table 3. Fine-grained results for WN18.

	All (100%)		**Sym** (21.4%)		**Eq** (72.5%)		**UC** (6.1%)	
	hits@1	hits@10	hits@1	hits@10	hits@1	hits@10	hits@1	hits@10
AMIE	.872	.948	**1.000**	**1.0**	.904	**1.0**	.047	.166
RuleN	**.945**	**.958**	.999	**1.0**	**.998**	**1.0**	**.128**	**.325**
HolE	.933	.940	.981	**1.0**	**.998**	.999	.011	.039
RESCAL	.749	.874	.878	.973	.772	.913	.019	.063
TransE	.082	.944	.000	.988	.114	.996	.000	.175

Table 4. Fine-grained results for FB15k (h@k refers to hits@k).

	All (100%)		**Sym** (7.2%)		**Eq** (60%)		**Sub** (6.8%)		$\mathbf{P_2}$ (7.3%)		**UC** (18.4%)	
	h@1	h@10	h@1	h@10	h@1	h@10	h@1	h@10	h@1	h@10	h@1	h@10
AMIE	.647	.858	.906	.983	.766	.961	.720	.950	.451	**.736**	.205	**.486**
RuleN	**.772**	**.870**	**.992**	**1.0**	**.940**	**.982**	**.831**	**.954**	**.536**	.724	**.207**	.480
HolE	.366	.706	.046	.936	.484	.811	.505	.814	.179	.438	.127	.339
RESCAL	.267	.600	.126	.768	.308	.638	.333	.645	.288	.546	.158	.416
TransE	.031	.796	.000	.852	.039	.893	.024	.884	.019	.661	.027	.479

On FB15k-237, AMIE and RuleN outperformed the other approaches only in the P_2 category. The overall results were slightly below the best performing embedding-based systems RESCAL and TransE as they were superior on the large *Uncovered* subset. These different strengths indicate potential for an ensemble model.

To sum up, some of the approaches that are based on embeddings had rather specific problems with symmetric relations in our experiments. Furthermore, the other subsets that can be covered by highly confident path rules of length one or two, could not be solved reliably by approaches such as TransE, HolE, or RESCAL. This became more obvious when looking at hits@1 instead of looking at hits@10. Overall, we observed rule-based approaches to be more precise. Their top ranked candidate was usually a correct hit (for most categories $>$50%), while this was not the case for TransE, HolE, or RESCAL. On the other hand, those systems held their ground in test cases that are tough for rule-based systems.

4.4 Ensemble Learning

Given that rule-based and embedding-based approaches use unrelated strategies and therefore achieve different results on specific categories, we propose to combine both methods to produce predictions with higher quality. The training time of an ensemble is essentially bottlenecked by the system that requires the most computational effort since models can be built in parallel. Learning the ensemble weights is a negligible effort in comparison. Hence, we feel that this approach is practical given sufficient resources.

We constructed an ensemble that consists of RuleN and AMIE on the one hand and TransE, HolE and RESCAL on the other hand using linear blending to combine these models, as suggested in [15]. The goal is to combine the strength of each model at the

Table 5. Fine-grained results for FB15k-237.

	All (100%)		$\mathbf{P}_2$ (14%)		**UC** (86%)	
	hits@1	hits@10	hits@1	hits@10	hits@1	hits@10
AMIE	.174	.409	.437	.656	.131	.368
RuleN	**.182**	.420	**.487**	**.691**	.132	.376
HolE	.096	.291	.166	.337	.085	.283
RESCAL	.167	.418	.342	.546	**.138**	.397
TransE	.106	**.430**	.191	.579	.092	**.405**

relation level. This is in line with our observation that there are relations for which RuleN or AMIE can learn rules with high confidence, while there are also relations where it is not possible to learn such rules. We constructed for each relation a dataset that consisted of all its triples from the training set as well as an equal amount of negative triples obtained by randomly perturbing either subject or object. Then a meta learner (logistic regression) was trained such that the constructed data could be classified correctly, using each individual model's normalized score as input feature.

Learning the weights based on the performance on the training set has its drawbacks. Rule-based systems need access to the training set to infer new knowledge from learned rules. Given this fact, they could trivially replicate all knowledge contained in the training set. To prevent this for each completion task, the triple that defines this task needs to be temporarily suppressed. Embedding-based systems, on the other hand, are trained with the primary goal of remembering the training set as good as possible. To establish equal preconditions, a similar tweak would have to be applied to these systems. However, it is impractical to do so given their latent knowledge representation. Learning the ensemble weights on the validation set, i.e., performing link prediction on unseen data, might be a better alternative. However, in most of the existing works the validation set was used for hyperparameter tuning only. Thus, we refrained from doing so to prevent doubts about the comparability of the results.

Instead of presenting results in terms of filtered hits@k with a fixed k, we visualized hits@k for $k = 1 \dots 50$ in Figure 4. At the bottom, we also added the filtered mean reciprocal rank (MRR).[2] The performance gain of the ensemble over its best performing member system varied between the different datasets. For WN18, it achieved slightly inferior results than the best single approach, which is RuleN. We cannot fully explain the small loss of quality of the ensemble. It should be noted that the characteristics of WN18 heavily reward rule-based systems and that this might be an example for the problem described in the previous paragraph. On FB15k, the ensemble was clearly better than the best single approach, which was again RuleN. The results of the ensemble were about 3 percentage points better over the whole range of k. The ensemble was even more beneficial on FB15k-237. This supports our assumption that the performance gain of the ensemble over its rule-based member systems correlates with the size of the *Uncovered* fraction of a data set. The high precision of rule-based systems is reflected

[2] If a rule-based approach did not rank the candidate, we have set the rank to $n/2$ where n is the set of all entities. This is the average result of randomly ranking the candidates.

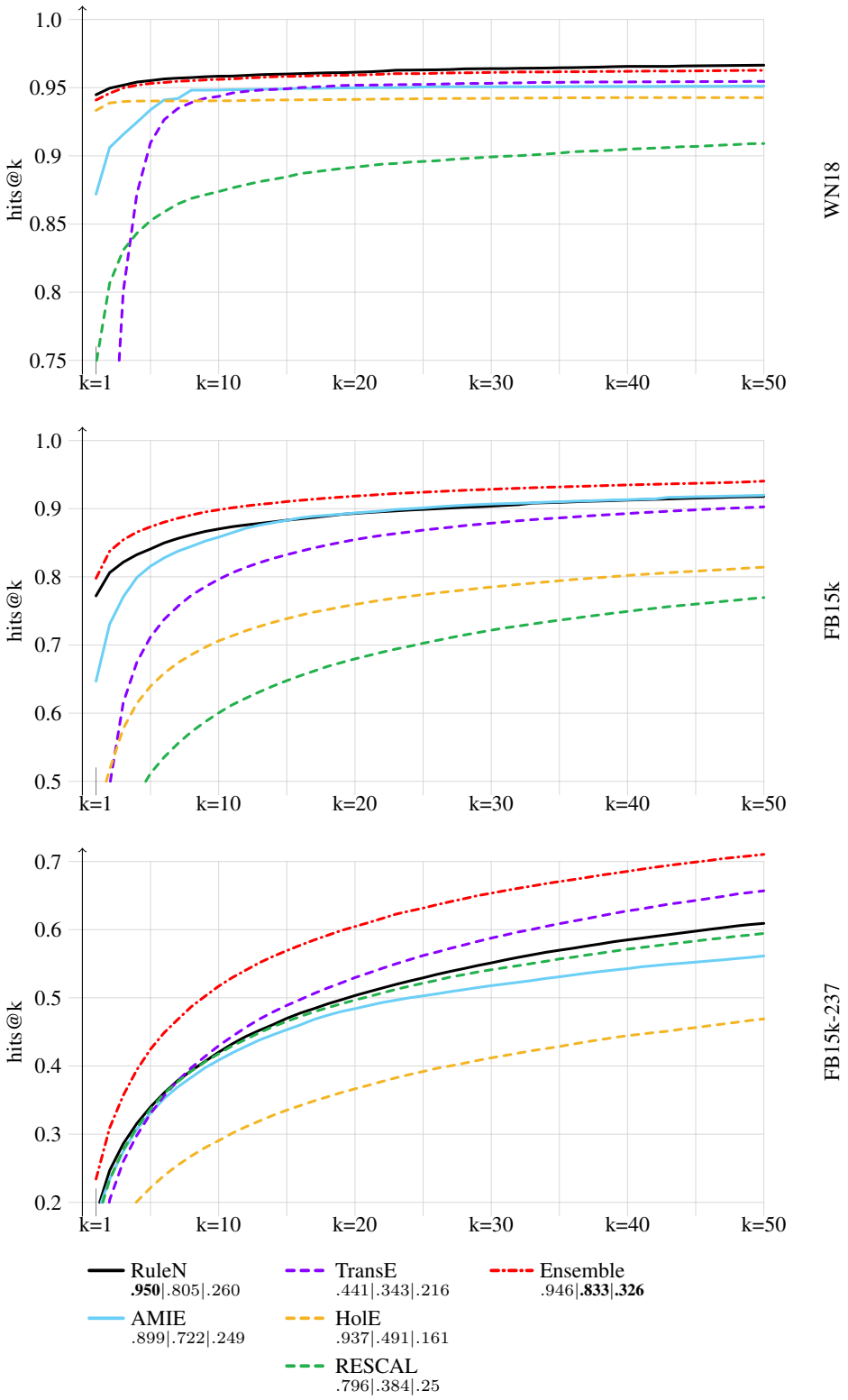

Fig. 4. Hits@k for k=1 . . . 50 for different systems and ensembles for WN18, FB15k and FB15k-237. Filtered MRRs are shown below the explanation for each approach in the order WN18|FB15k|FB15k-237.

both in the hits@1 score and the MRR. With the exception of WN18, these scores are further improved by the ensemble.

Additionally, we have analyzed the ensemble weights that have been learned for FB15k-237. The relation *nationality* is an example for which RuleN has high weights. For this relation, RuleN generates many C rules, which reflect the frequency distribution of the different nationalities (most people are from the US, followed by UK, and so on). We have also checked other examples of high weights for rule-based approaches. Most of them were correlated with the existence of rules with high confidence.

The results of our ensemble support the idea that embedding- and rule-based approaches perform well on different types of completion tasks, and that it is fruitful to join predictions of both types of models. This is especially important for datasets that might have less regularities than the datasets usually used for evaluation purposes. For such datasets a combination of both families might be even more beneficial.

5 Conclusion

In this paper, we analyzed rule-based systems for knowledge graph completion on datasets commonly used to evaluate embedding-based models. The generated results allow for a comparison with embedding-based approaches for this task. Besides global measures to rank the different methods, we also classified test cases of the datasets based on the explanations generated by our rule-based approach. This partitioning is available for future works. We gained several interesting insights.

- Both AMIE and RuleN are for the most commonly used datasets competitive to embedding-based approaches. This holds not only with respect to TransE, RESCAL, or HolE, but still holds for the large majority of the models reported about in [13] and [6]. Only few of these embedding models perform slightly better.
- Rule-based approaches can deliver an explanation for the generated ranking. This feature can be used for a fine-grained evaluation and helps to understand the regularities within and the hardness of a dataset.
- TransE, RESCAL, and HolE have problems in solving specific types of completion tasks that can be solved easily with rule-based approaches. This becomes noticeable in particular when looking solely at the top candidate of the filtered ranking.
- The good results of the rule-based systems are caused by the fact that the standard datasets are dominated by regularities such as symmetry and (inverse) equivalence. FB15k-237 is an exception to this due to the specific way it was constructed.
- It is possible to leverage the outcome of both families of approaches by learning an ensemble. This ensemble achieves better results than any of its members (the WN18 results are a minor deviation).

With this paper, we tried to fill a research gap and shed new light on the insights gained in previous years. Rule-based approaches perform very well and are a competitive alternative to models based on embeddings. For that reason, they should be included as a baseline for the evaluation of knowledge graph completion methods. Moreover, we recommend conducting the evaluation on a more fine-grained level like the one we proposed.

References

1. Bordes, A., Glorot, X., Weston, J., Bengio, Y.: A semantic matching energy function for learning with multi-relational data. Machine Learning 94(2), 233–259 (2014)
2. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in neural information processing systems. pp. 2787–2795 (2013)
3. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2d knowledge graph embeddings. CoRR abs/1707.01476 (2017), http://arxiv.org/abs/1707.01476
4. Galárraga, L.A., Teflioudi, C., Hose, K., Suchanek, F.: Amie: association rule mining under incomplete evidence in ontological knowledge bases. In: Proceedings of the 22nd international conference on World Wide Web. pp. 413–422. ACM (2013)
5. Gardner, M., Mitchell, T.M.: Efficient and expressive knowledge base completion using subgraph feature extraction. In: EMNLP. pp. 1488–1498 (2015)
6. Kadlec, R., Bajgar, O., Kleindienst, J.: Knowledge base completion: Baselines strike back. arXiv preprint arXiv:1705.10744 (2017)
7. Lao, N., Mitchell, T., Cohen, W.W.: Random walk inference and learning in a large scale knowledge base. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 529–539. Association for Computational Linguistics (2011)
8. Nickel, M., Rosasco, L., Poggio, T.A., et al.: Holographic embeddings of knowledge graphs. In: AAAI. pp. 1955–1961 (2016)
9. Nickel, M., Tresp, V., Kriegel, H.P.: A three-way model for collective learning on multi-relational data. In: ICML. vol. 11, pp. 809–816 (2011)
10. Niepert, M.: Discriminative gaifman models. In: Advances in Neural Information Processing Systems. pp. 3405–3413 (2016)
11. Schlichtkrull, M., Kipf, T.N., Bloem, P., Berg, R.v.d., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. arXiv preprint arXiv:1703.06103 (2017)
12. Shen, Y., Huang, P.S., Chang, M.W., Gao, J.: Traversing knowledge graph in vector space without symbolic space guidance. arXiv preprint arXiv:1611.04642 (2016)
13. Shi, B., Weninger, T.: Proje: Embedding projection for knowledge graph completion. In: AAAI. vol. 17, pp. 1236–1242 (2017)
14. Toutanova, K., Chen, D.: Observed versus latent features for knowledge base and text inference. In: Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality. pp. 57–66 (2015)
15. Wang, Y., Gemulla, R., Li, H.: On multi-relational link prediction with bilinear models. arXiv preprint arXiv:1709.04808 (2017)
16. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: AAAI. vol. 14, pp. 1112–1119 (2014)
17. Xiao, H., Huang, M., Zhu, X.: Transg: A generative model for knowledge graph embedding. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). vol. 1, pp. 2316–2325 (2016)
18. Yang, B., Yih, W.t., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint arXiv:1412.6575 (2014)
19. Zeng, Q., Patel, J.M., Page, D.: Quickfoil: Scalable inductive logic programming. Proceedings of the VLDB Endowment 8(3), 197–208 (2014)