Numerical Literals in Link Prediction: A Critical Examination of Models and Datasets

Moritz Blum[1(✉)], Basil Ell[1,2(✉)], Hannes Ill[1(✉)], and Philipp Cimiano[1(✉)]

[1] Bielefeld University, CITEC, Inspiration 1, 33619 Bielefeld, Germany
{mblum,bell,cimiano}@techfak.uni-bielefeld.de, hannes.ill@tum.de
[2] University of Oslo, Problemveien 11, 0313 Oslo, Norway
basile@ifi.uio.no

Abstract. Link Prediction (LP) is an essential task over Knowledge Graphs (KGs), traditionally focussed on using and predicting the relations between entities. Textual entity descriptions have already been shown to be valuable, but models that incorporate numerical literals have shown minor improvements on existing benchmark datasets. It is unclear whether a model is actually better in using numerical literals, or better capable of utilizing the graph structure. This raises doubts about the effectiveness of these methods and about the suitability of the existing benchmark datasets.

We propose a methodology to evaluate LP models that incorporate numerical literals. We propose i) a new synthetic dataset to better understand how well these models use numerical literals and ii) dataset ablations strategies to investigate potential difficulties with the existing datasets. We identify a prevalent trend: many models underutilize literal information and potentially rely on additional parameters for performance gains. Our investigation highlights the need for more extensive evaluations when releasing new models and datasets.

Keywords: Link Prediction · Numerical Literals · Evaluation

1 Introduction

Knowledge Graphs (KGs) store information in a graph-structured form as sets of relational triples (i. e., triples with a relation that connects two entities) and attributive triples (i. e., triples with a relation that annotates an entity with literal information). Prominent KGs are *Freebase* [5] and *Wikidata* [28]. A small example KG is shown in Fig. 1. KGs have emerged as a method to represent and store knowledge in various domains and applications, and will, as the authors believe, play an important role in generative AI as they complement LLMs for Retrieval Augmented Generation [16]. Nevertheless, KGs are inherently incomplete for various reasons [7]. In the past, efforts have been made to develop Link

G. Demartini et al. (Eds.): ISWC 2024, LNCS 15231, pp. 23–46, 2025.
https://doi.org/10.1007/978-3-031-77844-5_2

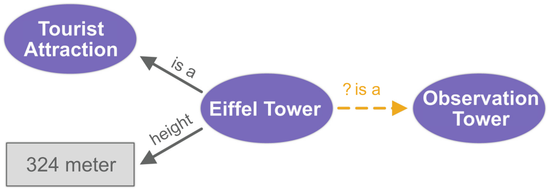


Fig. 1. Example KG about the *Eiffel Tower*. The KG contains the entities *Eiffel Tower*, *Tourist Attraction*, and *Observation Tower*; and the literal value *324 m*: the height of the *Eiffel Tower*.

Prediction (LP) methods to predict missing triples based on the triples already available.

Most LP approaches focus on relational triples, i. e., only use relational triples to predict missing relational triples. Neither do these models make use of attributive triples when predicting relational triples, nor do these models predict attributive triples – these models ignore information encoded in literals that might be valuable. In the example shown in Fig. 1, a model should predict the *is a* relation between *Eiffel Tower* and *Observation Tower*. Ideally, a model has learned from the data that a *Tourist Attraction* that has a certain height (e.g., above some threshold) is an *Observation Tower*. Here, a model needs to incorporate information expressed by the relational triple (*Eiffel Tower*, *is a*, *Tourist Attraction*) and information expressed by the attributive triple (*Eiffel Tower*, *height*, *324 m*) to predict the missing triple.

To incorporate literals, specialized models [30,34] or extensions of models were proposed [3,8,13,24,31]. Most models only operate on one and only some on multiple types of literals. Even though some models are technically able to predict literal values (e. g., [17,31,32]), we focus on the more prominent task of incorporating literals into the prediction of relational triples. Language Models (LMs) that operate on textual entity descriptions recently set state-of-the-art performance for LP [20]. The inclusion of numerical literals has shown only small improvements in models that do not use literals on common benchmark datasets [13,31]. However, we assume that numerical literals are highly valuable, especially for scientific KGs about physical experiments [4] or manufacturing processes [23], which store a large amount of information as numerical data. We focus on numerical literals as there is a lack of research on how well LP models that were designed to be able to use numerical literals can and do make use of numerical literals.

Typically, when new models or model extensions that can incorporate literals are published, then they are compared to state-of-the-art models that can incorporate numerical literals and to the base model they extend by standard metrics like Mean Reciprocal Rank (MRR) on benchmark datasets. As the improvements through incorporating literals is minor, we can not be sure whether the models are using the attributive triples, or whether the attributive triples in the existing datasets are valuable.

The benchmark datasets that contain literals are often created by enriching standard LP datasets, e. g., FB15k-237 or YAGO3-10, with literal information from their larger source KGs. These datasets might not be perfectly suited for the evaluation of LP models that incorporate literals, as, e. g., a certain amount of attributive triples in FB15k-237 connect entities to identifiers (IDs) for other databases that are not exploitable by a model.[1] Here, we only point to the ID relations, but other attributive triples might also not be valuable. Something similar might be the case for most of these datasets. To the best of our knowledge, no published evaluation has proven that the numerical literals in these datasets provide information relevant for LP. Therefore, we can not investigate whether the literals are used by the models, making them not suitable for benchmarking.

Overall, research on LP with numerical literals lacks a detailed evaluation of and comparison with the existing models and lacks insights on the used benchmark datasets. Therefore, we present the following contributions: i) We propose a method to extend a dataset with relational and attributive triples, where the prediction of relational triples of that kind can only then be carried out successfully if a model makes use of the attributive triples. ii) We propose ablation strategies for the existing evaluation datasets to investigate whether the numerical literals provide any additional knowledge, or whether the numerical literals only add information already contained in the relational triples. iii) We evaluate existing LP models which state to incorporate numerical literals on our semi-synthetic benchmark dataset and on datasets that we obtained by applying our ablation strategies to existing benchmark datasets, to gain insights into the models capabilities to incorporate literals and to gain insights into the suitability and difficulty of the existing datasets.

2 Preliminaries

With G we denote a directed labeled multi-graph with numerical literals. G is a set of triples $(s, p, o) \in \mathcal{U} \times \mathcal{U} \times (\mathcal{U} \cup \mathbb{R})$, where $\mathcal{U}$, and $\mathbb{R}$ are disjoint sets of URIs and numerical values.[2] The set of triples can be categorized into relational triples G_E and attributive triples G_A:

$$G_E = \{(s, p, o) \mid \exists (s, p, o) \in G \;\; \text{s.t.} \;\; o \in \mathcal{U}\} \tag{1}$$

$$G_A = \{(s, p, v) \mid \exists (s, p, v) \in G \;\; \text{s.t.} \;\; v \in \mathbb{R}\} \tag{2}$$

[1] Table 3 in Appendix A shows that for FB15k-237, 3/10 of the triples related to the most frequent attributive relations hold such IDs, overall 6.9% of all attributive triples in the dataset.

[2] Although typically KGs contain various types of literals, such as string literals or date literals, here we focus only on numerical literals and we do not distinguish between different types of numerical literals such as integer and float.

The set of entities $\mathcal{E} \subseteq \mathcal{U}$, the set of entity relations $\mathcal{R}_E \subseteq \mathcal{U}$, and the set of attributive relations $\mathcal{R}_A \subseteq \mathcal{U}$ are defined as follows:

$$\mathcal{E} = \{x \mid \exists(s,p,o) \in G \ \text{ s.t. } \ x{=}s \vee (x{=}o \wedge o \in \mathcal{U})\} \tag{3}$$

$$\mathcal{R}_E = \{p \mid \exists(s,p,o) \in G_E\} \tag{4}$$

$$\mathcal{R}_A = \{p \mid \exists(s,p,v) \in G_A\} \tag{5}$$

2.1 Link Prediction Models

LP models are trained to predict missing triples using triples already available.

Traditional Link Prediction Models. Traditional LP models can be considered as a function f that assigns a score $f(\vec{s},\vec{p},\vec{o}) \in \mathbb{R}$ to each triple (s,p,o) where $s,p,o \in \mathcal{U}$ and $\vec{e}$ denotes the embedding of the entity e. These models are trained to score true triples (i. e., triples in $\mathcal{G}_E$) higher than false triples (i. e., triples not in $\mathcal{G}_E$). Notably, the conventional LP models do not incorporate literals. Popular models are TransE [6], DistMult [33], ComplEx [27], and TuckER [2].

Link Prediction Approaches Incorporating Numerical Literals. Some LP models that are able to incorporate numerical literals are extensions of traditional LP models. These models use a feature vector $\vec{x}_e$ for each entity $e \in \mathcal{E}$. Each dimension of the feature vector $\vec{x}_e$ corresponds to a relation $r \in \mathcal{R}_A$. The value for a dimension is randomly selected from $\{v \mid \exists(e,r,v) \in G_A\}$ or is set to *"0"*, when the entity e has no value for the relation r in G_A.[3] These models can be categorized into two types.

Fusion via a Modification of the Scoring Function. Such models use the numerical features $\vec{x}_e$ as (additional) input features, i. e., they modify the scoring function explicitly. We investigate two established approaches: i) *LiteralE* [13], by Kristiadi et al., extends traditional LP models by adding a learnable parametric gate function $g(\vec{e},\vec{x}_e)$ to obtain a literal-enriched entity embedding that replaces the initial embedding in the scoring function. This makes LiteralE universally combinable with most existing embedding methods. In this paper, we evaluate LiteralE$_{DistMult}$ and LiteralE$_{ComplEx}$. ii) *KBLN* [8] is a reduced variant of KBLRN by García-Durán et al. KBLRN is a product of experts model that combines relational (vectors that describe in which graph patterns an entity occurs), latent (entity and relation embeddings), and numerical literal features. KBLN leaves out the expert for relational information.

[3] Replacing non-existing features with *"0"* can be considered critical as *"0"* might also be a valid literal value. Established methods like, e. g., LiteralE, make this abstraction. To ensure no negative effect, we computed the proportion of *"0"* literal values in the used datasets which is marginally small: FB15k-237 0.006%, YAGO3-10 0%, LitWD48K 1.67%.

Fusion via a Modification of the Objective Function. Such models learn to predict numerical features jointly with the LP objective.[4] Thereby, the entity embeddings incorporate information from both the graph structure and numerical literals. In this paper, we investigate two established approaches: i) *MTKGNN* [25], by Tay et al., introduces a neural network for numerical value regression in addition to a neural network for triple scoring. ii) *TransEA* [31], by Wu et al., is an extension of TransE [6] that learns a set of functions $g = \{g_p \mid p \in \mathcal{R}_A\}$ for numerical value regression.

A different line of research investigates methods applied to the datasets instead of the models.

Fusion via Literal Transformations. Models that transform attributive triples into relational triples allow traditional LP models to incorporate literal information without modifying the scoring function or objective function. In this paper, we investigate the following approach considered state-of-the-art in LP with numerical literals: *KGA* [29], by Wang et al., transforms numerical attributive triples into relational triples by discretizing numerical values into bins, and chaining these bins, modeling multiple levels of granularity.

For a broader overview of literal-aware LP models, we refer to [10].

2.2 Datasets

Widely used LP datasets that contain numerical literals are: FB15k-237 [26], YAGO3-10 [15], and LitWD48K [9]. An extensive overview about existing datasets and their types of literals can be found in [9].

We briefly describe the datasets used in this paper. These datasets are publicly accessible under *CC-BY* licenses. i) *FB15k-237* is a subset of FB15k which is a subset of Freebase. Toutanova et al. created FB15k-237 by removing inverse relations from FB15k that allowed even simple models to achieve high scores by simply inverting triples [26]. We use the version of FB15k-237 that was extended with numerical literal as provided by Kristiadi et al. [13].[5] ii) *YAGO3-10* is a subset of *YAGO3* [14] that only contains triples associated with entities that occur in at least ten relations leading mostly to triples related to people. YAGO3-10 does not contain literals, but they can be derived from *YAGO3*. Again, we use the numerical literals provided by Kristiadi et al. iii) *LiterallyWikidata* [9] comprises three datasets designed to evaluate LP models utilizing literal data, sourced from *Wikidata* and *Wikipedia*. These datasets vary in size and structure; we use the largest, LitWD48K.

Table 2 in Appendix A shows the general characteristics of the datasets used in this paper.

---

[4] The literal information is implicitly encoded into the embeddings and not explicitly provided during inference.

[5] See https://github.com/SmartDataAnalytics/LiteralE/tree/master/data.

2.3 Evaluation Metrics

The models are compared via the filtered mean rank (MR) metric, the mean reciprocal rank (MRR) metric, and Hits@k for $k \in \{1, 3, 10\}$, as proposed by [6]. For each triple in the test set, the subject and the object entities are corrupted by replacing them by any $e \in \mathcal{E}$. The score for each triple is used to rank the test triple among all of those triples by sorting in ascending order. Triples already contained in the graph are removed before ranking to not cause true triples to increase the rank of the test triples.

The MR is the mean rank, the MRR is the mean of the multiplicative inverse of the ranks, and Hits@k is the proportion of ranks $\leq k$.

3 Related Work

To the best of our knowledge, no existing work focuses on a methodology for evaluating LP models with numerical literals or other types of literals.

LP models are evaluated according to standard metrics such as *MR* and *MRR*, following the evaluation protocol proposed by Bordes [6], where models are treated as black-boxes. Safavi et al. raise concerns about the reliability of these ranking-based metrics [19]. They point out that while a ranking metric may suggest good performance because the correct triple is ranked high, it could still receive a significantly lower score than an incorrectly top-ranked triple.

In-depth evaluations, e. g., analyzing particular relations or distinguishing between head and tail predictions (as done by Bordes et al. [6]), are uncommon. Such an analysis can be useful to investigate how specific KG characteristics can be learned by a model, e. g., if symmetric relations can be properly represented by the model.

Explainability methods could provide insights into the behavior of LP models. Whereas rule-based approaches [12] and explainers over graph neural network-based approaches [35,36] can offer explanations for model predictions, particularly shallow models like TransE or DistMult are difficult to explain. Ismaeil et al. generate interpretable vectors for entity embeddings [11]. They employ embedded feature selection techniques to extract propositional features from the KG that are important for a given KG embedding model.

Another way to gain insights into the specific behavior and capabilities of LP models is to build datasets in a way such that obtaining good LP results requires the models to have specific capabilities, thus these datasets enable to test to what extent a model has these capabilities. E. g., recent work stated a lack of evaluation datasets covering certain KG properties like a given entity type system, pairs of mutually inverse relations, or mediator objects to represent n-ary relationships. Shirvani-Mahdavi et al. evaluate these properties on a newly proposed version of the Freebase KG [22].

The outlined open challenge of evaluating and explaining traditional LP models without considering literals results in a scarcity of research on the evaluation and explainability of LP with literals.

Although the original releases of existing LP datasets such as FB15k-237 and YAGO3-10 lack attributive triples, they have been extended with textual and numerical attributes sourced from their respective KGs. However, these enriched datasets contain numerous entities without numerical attributes,[6] and the provided numerical attributes are not proven to be helpful for LP. Consequently, Gesese et al. introduced a series of LP datasets called LiterallyWikidata, constructed from Wikidata and Wikipedia, specifically for LP involving numerical and textual literals [9]. The graph structure of LiterallyWikidata was designed for benchmarking LP models, avoiding issues such as that inverse relations could leak information or the existence of any shortcut features.

Despite the existence of datasets tailored for LP tasks involving (numerical) literals, these datasets are derived from real KGs, making it challenging to accurately assess the true advantages of integrating (numerical) literal information.

The most related work is García-Durán et al.'s input feature ablation study, which investigated which graph structure features improved their model's performance [8].

4 Methodology

We i) propose a method to enrich an existing dataset with synthetic information that enables us to find out if numerical literal-aware models are capable of using numerical literals to make predictions about relational triples. Furthermore, ii) we develop a set of ablation methods to gain further insights into the existing literal-aware datasets, whether in some datasets attributive triples might not be used for LP, or whether information is represented redundantly as attributive and relational triples.

When elucidating the derivable conclusions from the following ablation experiments, we denote a model trained on the dataset D as $m(D)$ and define $\sigma(m(D))$ as the result of evaluating $m(D)$ according to some measure of performance σ (such that a higher value indicates better performance).

4.1 Semi-synthetic LP Dataset with Literals

To ensure attributive triples to be relevant, we propose a dataset extension methodology with the intention to introduce a new learning goal given by a function h into the dataset.

For simplicity, we restrict h to be a function that predicts a relation r_{syn-r} from an existing entity e to one of two entities representing classes, added to the dataset, namely c_{high} and c_{low}, based on the attributive triple (e, r_{syn-a}, v). More precisely, our function h is defined as:

$$h(e) = \begin{cases} (e, r_{\text{syn-}r}, c_{\text{high}}) & \text{if } \exists (e, r_{\text{syn-}a}, v) \in G' \quad \text{with } v > 0.5 \\ (e, r_{\text{syn-}r}, c_{\text{low}}) & \text{if } \exists (e, r_{\text{syn-}a}, v) \in G' \quad \text{with } v \leq 0.5 \end{cases}$$

[6] See Table 2 in Appendix A.

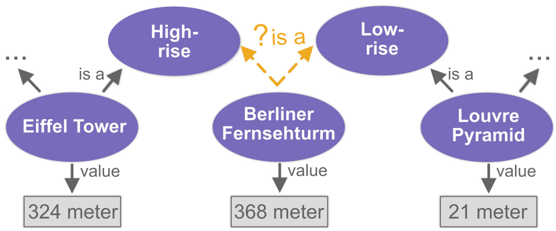


Fig. 2. Example of the synthetic dataset enrichment. The entitites *High-rise* and *Low-rise* represent c_{high} and c_{low} and *is a* is used as the r_{syn-a} relation. Ideally, an LP model predicts the tail entity *High-rise* for the given head *Berliner Fernsehturm* and the *is a* relation.

To remove any noise and biases from the original G_A, we replace it by G'_A defined as $\{(e, r_{syn-a}, v) \mid e \in \mathcal{E}'\}$ where $v \sim \text{Uniform}(0,1)$ and $\mathcal{E}' \subseteq \mathcal{E}$. We then apply h to every $e \in \mathcal{E}'$ to obtain relational triples that are added to G_E to create G'_E. The new dataset is defined as $G' = G'_E \cup \mathcal{G}'_A$. An example is shown in Fig. 2.

Note that the function h could be more complex, taking into account multiple relational and attributive triples, make use of more than the one target relation r_{syn-r} and the two target entities c_{high} and c_{low}, and realize something more complex than comparing a value against a threshold value.

Let $\mathcal{E}_{high}$ and $\mathcal{E}_{low}$ be defined as follows: $\mathcal{E}_{high} := \{e \in \mathcal{E}' \mid \exists (e, r_{syn-a}, v) \in G'_A \wedge v > 0.5\}$ and $\mathcal{E}_{low} := \{e \in \mathcal{E}' \mid \exists (e, r_{syn-a}, v) \in G'_A \wedge v \leq 0.5\}$. (Note that $\mathcal{E} = \mathcal{E}_{high} \cup \mathcal{E}_{low}$ and $\mathcal{E}_{high} \cap \mathcal{E}_{low} = \varnothing$.)

Our goal is to measure the models' ability to score the synthetic relational triples according to h. As this is a binary classification task, we define the accuracy, denoted by Acc, as follows:

$$Acc := \frac{true_{high} + true_{low}}{|\mathcal{E}_{high}| + |\mathcal{E}_{low}|} \tag{6}$$

where $true_{high}$ is the number of $e \in \mathcal{E}_{heigh}$ for which $r(e, r_{syn-a}, c_{high}) \geq r(e, r_{syn-a}, c_{low})$. $true_{low}$ is defined analogously.

We consider the following situations where we can derive conclusions:

i) if $\sigma(m(G_E \cup G'_A)) < \sigma(m(G_E \cup G_A))$, i.e., the model that has no access to the original attributive triples performs worse, then the attributive triples are used for the prediction;
ii) if $\sigma(m(G_E \cup G'_A)) \geq \sigma(m(G_E \cup G_A))$, i.e., both models perform equally well, or the model that used the random features performs better, then the model is not capable of making use of literals.

4.2 Literal Features Ablation

If a model has proven to incorporate literals from a synthetic dataset into the prediction, this model should be evaluated on the established benchmark datasets.

To gain insights into the performance increase by using literals, one has to compare models that use literals against the same model without access to literals.

For some models, e.g., MTKGNN, we cannot remove the attributive triples from the dataset, as these models directly operate on the numerical features only, and do not learn a separate entity embedding. For other models, e. g., LiteralE, removing the attributive triples from the dataset reduces the number of model parameters.

Therefore, we propose an ablation method where each entity is related with each attributive relation to a certain value. Given $G = G_E \cup G_A$, we create $G' = G_E \cup G'_A$ where G'_A is created as follows: for each entity $e \in \mathcal{E}$ and each relation $p \in \mathcal{R}_A$, we add the triple (e, p, v) to G'_A where v is a certain literal value we assign.

As some models only operate on numerical features, assigning the same value to all attributive triples would lead to identical features for all entities. Consequently, such models would lose the ability to distinguish the entities. Therefore, we propose to sample v randomly from Uniform$(0, 1)$.

We consider the following situations where we can derive conclusions about a model and a dataset, under the assumption that the model can make use of literals according to the experiments of the model on the semi-synthetic dataset:

i) if $\sigma(m(G_E \cup G'_A)) < \sigma(m(G_E \cup G_A))$, i. e., the model that has no access to the original attributive triples performs worse, then the attributive triples are relevant for the prediction task;
ii) if $\sigma(m(G_E \cup G'_A)) \geq \sigma(m(G_E \cup G_A))$, i. e., both models perform equally well, or the model that used the random features performs better, then, either the information represented via attributive triples is redundantly represented via relational triples, the literal information is difficult to use by the models, or no information relevant for LP is represented via attributive triples.

4.3 Relational Features Ablation

The previous experiments may leave open whether attributive triples are not relevant for the prediction task, challenging to leverage, or whether information is represented redundantly as relational and attributive triples. To gain insights into the redundancy of relational and attributive triples for a given dataset, we propose an ablation method that targets relational triples, thus modifies G_E. We reduce G_E to $G_{E_{-\alpha}}$ s. t.

1) $|G_{E_{-\alpha}}| = (1 - \alpha)|G_E|$ where $\alpha \in [0, 1]$ is a user-defined real value.
2) $\forall e \in \mathcal{E} : (\exists p, o : (e, p, o) \in G_{E_{-\alpha}}) \vee (\exists s, p : (s, p, o) \in G_{E_{-\alpha}})$
3) $\forall p \in \mathcal{R}_{\mathcal{E}} : (\exists s, o : (s, p, o) \in G_{E_{-\alpha}})$

This means, we remove relational triples from G until $|G_{E_{-\alpha}}| = (1 - \alpha)|G_E|$. We ensure that there remains at least one triple per entity $e \in \mathcal{E}$ and relation $p \in \mathcal{R}_{\mathcal{E}}$ such that embeddings are learned. Note that for some G_E and $\alpha \in \mathbb{R}^+$ it can be the case that there is no $G_{E_{-\alpha}}$ that satisfies both constraints. Thus, there is a limit to how much G_E can be reduced.

We consider the following situations where we can derive conclusions:

i) if with the reduced set of relational triples the random feature ablation has an effect on model performance (i.e., $\sigma(m(G_{E_{-\alpha}} \cup G_A)) \gg \sigma(m(G_{E_{-\alpha}} \cup G'_A)))$, then that means that information is represented redundantly as attributive and relational triples and that attributive triples are relevant for the prediction task;
ii) if with the reduced set of relational triples the random feature ablation has still no effect on the model performance (i.e., $\sigma(m(G_{E_{-\alpha}} \cup G_A)) \approx \sigma(m(G_{E_{-\alpha}} \cup G'_A)))$, then attributive triples are either difficult to incorporate or not relevant for the prediction task.

5 Experimental Setup

We apply our methodology to all numerical literal-aware models mentioned in [10] and the state-of-the-art model KGA [29].

Implementation. We run our experiments with LiteralE$_{DistMult}$, Literal E$_{ComplEx}$, KBLN, and MTKGNN with the code of Kristiadi et al. [13].[7]

We implemented TransEA in PyTorch Geometric[8] due to the absence of a public implementation.

We decided to use the model variants that achieve the overall best performance and the model that shows the largest performance gains through incorporating literals, which are KGA$_{TuckER}$ and KGA$_{DistMult}$ according to [29].[9]

All hyperparameters are reported in Appendix B. We ran all experiments three times and computed mean and standard deviation for each metric.

Semi-Synthetic FB15k-237. We apply our dataset enrichment method to the FB15k-237 dataset. We decided that $\mathcal{E}$ is the set of entities of type person. FB15k-237 contains $4,505$ entities of type person, i. e. ≈30% of the entities.[10]

For evaluation, we create a training, validation, and test split as follows: we add 70% of the new synthetic relational triples to the original train set, 15% to the original validation set, and 15% to the original test set. The new synthetic literal values replace the original literal values. The models are trained for LP as usual.

7 See https://github.com/SmartDataAnalytics/LiteralE.

8 See https://pytorch-geometric.readthedocs.io. We extended the existing TransE implementation to TransEA.

9 The KGA transformations approximately create as many additional relational triples as attributive triples. As our proposed attributive features ablation creates a large number of literals with random values, the number of relational triples increases significantly. Consequently, we had to limit the number of attributive relations. Instead of relating each entity with each attributive relation, we only replace the numerical values of attributive triples in the original dataset by a random value. Thereby, the model is provided with some literal information, i. e., the existence of the attributive relation.

10 By assigning numerical literals only to certain entities, we enable further analysis, such as determining if the model learns that only certain entities have a specific literal.

Computing Resources. Our evaluation required numerous experiments, due to the combinations of investigated models and datasets. We used 10 A100 GPUs for two weeks. The evaluated models have similar sizes, e. g., Literale$_{DistMult}$ trained on FB15k-237 has ≈3M parameters.

6 Results

6.1 Synthetic LP Dataset with Literals

We created a semi-synthetic FB15-237 dataset and used it to investigate the models' ability to utilize the necessary numerical literal information for predicting relational triples. The results are shown in Table 1. The accuracy for all models is shown in the column Acc_{org}. A score slightly above 0.5 suggests that the models' performance is only marginally better than random guess, and a score close to 1 suggests that the model is capable of making correct predictions by using the numerical literals.

As a baseline, we train the models with random features following the introduced literal features ablation method, which we applied after the creation of the semi-synthetic dataset. Acc_{rand} is the score of the models when the literals provide no information, forcing the models to guess randomly.

The variance across runs is small; hence, these values indicate a measure of reliable performance.

The KGA models are capable of using the provided numerical literals for their prediction as they achieve Acc_{org}'s of 0.999, both. The Acc$_{rand}$ scores range from 0.482 to 0.510, proving the models' random guessing. Note that due to the randomness of the features, the models cannot make a justified prediction. The Acc$_{org}$ scores achieved by the other models are much lower and in the same range as the Acc$_{rand}$ scores', showing that these models do not or do only to a small extent use the information provided via literals.

6.2 Literal Features Ablation

The results for our experiments for the random literal features ablation method are visualized for three models in Fig. 3 as box plots, showing the mean and variance of the MRR scores across three runs. For each combination of model and dataset, the plot shows two boxes. The first box shows the score achieved by the model trained with the original literal features and the second box shows the score achieved by the model trained on the random features.[11]

In general, as shown in Fig. 3, replacing original features by random features has no significant negative impact on most of the models as the box for the original features and the box for the random features overlaps in many cases and the differences are very small. The details reported in Appendix C show that in 9 of the 21 cases we see the models with random literals outperform

[11] The box-plots for all experiments and the scores of all computed metrics for these experiments are in Appendix C.

the models that use real literals regarding the MRR score, and only in 10 of the 15 cases models showed a benefit regarding the MRR score in using the real attributive triples provided by the datasets. The Hits@k scores follow this trend. When looking at the MR scores, this trend exists, too. The KGA models show the best usage of literals as the predictions with the original features are better, but only marginally, than with the random features for all combinations of KGA models and datasets investigated, except for $\text{KGA}_{DistMult}$ on the YAGO3-10 dataset.

Table 1. Scores achieved on the synthetic dataset. Acc_{org} denotes the Acc score achieved on the synthetic dataset when we provide the meaningful synthetic literal values, whereas Acc_{rand} denotes the Acc score on the synthetic dataset if we apply the random feature ablation after the dataset creation.

Model	$\textbf{Acc}_{org}$	$\textbf{Acc}_{rand}$
$\text{LiteralE}_{DistMult}$	0.512$\pm$0.003	0.482$\pm$0.001
$\text{LiteralE}_{ComplEx}$	0.493$\pm$0.005	0.498$\pm$0.020
KBLN	0.482$\pm$0.009	0.493$\pm$0.005
MTKGNN	0.472$\pm$0.006	0.495$\pm$0.009
TransEA	0.489$\pm$0.022	0.496$\pm$0.020
KGA_{TuckER}	0.999$\pm$0.000	0.510$\pm$0.007
$\text{KGA}_{DistMult}$	0.999$\pm$0.000	0.487$\pm$0.011

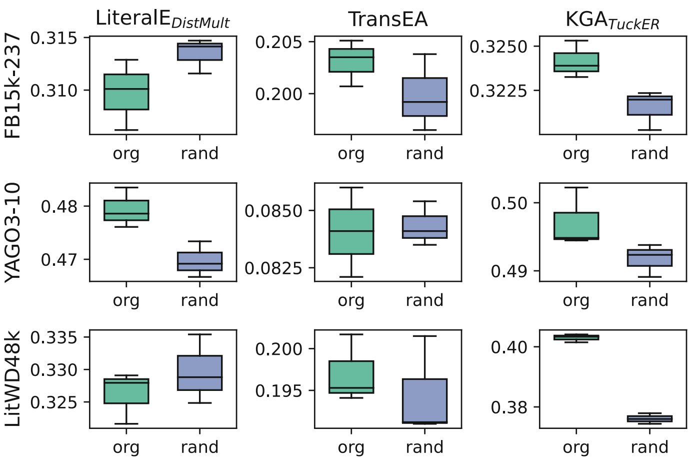


Fig. 3. MRR scores over three runs for models and datasets that either include the original literal features or that include random literal features.

6.3 Relational Features Ablation

The ablation effect of relational triples from FB15k-237 on KGA_{TuckER} is shown in Fig. 4.[12] We plot the mean and standard deviation of the MRR score while reducing the amount of relational triples from 100% (representing the original dataset) to 10% (equivalent to removing 90% of the relational triples) in steps of 10%. As expected, the MRR score decreases significantly when reducing the available relational triples, showing that important information is eliminated.

We compared these models against the ones we applied the random literal feature ablation strategy on. The curves of the models operating on random features are very close to the curves of the models operating on the original features, not showing any advantages of the original features when reducing the relational triples.

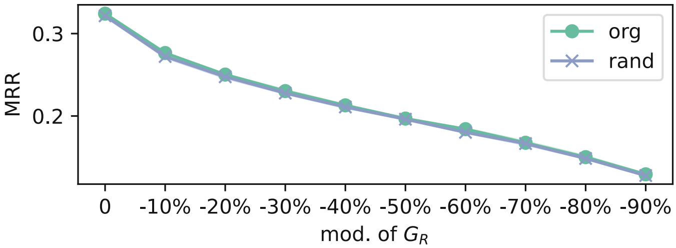


Fig. 4. KGA_{TuckER}'s MRR scores (mean and variance over three runs) after removing x% relational triples from FB15k-237. The model is provided either with the original or with random numerical features. The variance is marginally small and not visually recognizable in the figure.

7 Discussion

The synthetic dataset creates a scenario where numerical literals are necessary for predictions. KGA converts these continuous literals into discrete entities, allowing models to bypass the need to incorporate the concrete literal values. Thereby, KGA translates the task created by our semi-synthetic dataset into a more simple graph-structure learning task,[13] but might struggle with more complex synthetic datasets. The other models behaved similar to random guessing, as the scores are close to 0.5. We assume that these model's objective function does not enforce the models to integrate numerical information valuable for LP into the entity embeddings. Even though TransEA forces the embeddings to contain information to reconstruct the numerical literals, this information is not necessarily valuable for LP.

[12] Plots for further models are contained in Appendix C.

[13] Entities with similar literals obtain similar embeddings as entities with similar literals are connected to the same bins.

Even though we proposed our synthetic dataset to overcome a certain issue with the existing real-world benchmark datasets, we, nevertheless, believe that real-world datasets are relevant. Therefore, we also evaluated all models on these datasets and compared the scores to the scores of their random feature variant to confirm the previous results.

One would expect significant performance drops after applying the random feature ablation methods. We do not observe any significant performance drop, and in some cases even an improvement. Interestingly, the KGA models do only show small benefits of using numerical literals even though they showed good performances on our synthetic dataset. This brings us to the conclusion that either the literals are not relevant for the prediction task, or the information contained in attributive triples is difficult to use, or the information represented via the attributive triples is redundantly represented via relational triples.

We are not sure about the reason for the increase in performance after introducing random literals. Possibly, in some cases unintentionally good features are created which can be used by the models. A similar increase in performance through random node initialization has been observed by Abboud et al. for GNNs, which gain additional expressivity in the neighborhood encoding from random node initialization [1]. However, all models we investigated are shallow models that do not perform any neighborhood encoding.

We have to note that our scores of the LiteralE models slightly differ from the ones reported by Kristiadi et al. [13], even though we used their implementation and hyperparameters. Interestingly, the base ComplEx model achieves higher MRR and Hits@k scores than the LiteralE$_{ComplEx}$ model on FB15k-237 in our experiments. However, the similarity of the results from three runs confirm our results' reliability.

Lastly, we investigated if relational and attributive triples redundantly represent information, which leads to the LP performance to remain similar even though numerical literals are incorporated. If the attributive triples were redundant, one would expect the model with the real literal features to obtain less worse results when ablating the relational triples than the one with the random features, i. e., at some point the literal features should become important. As Fig. 4 does not show any benefit of incorporating numerical literals when reducing the amount of available relational triples, we conclude that either the attributive triples are not relevant for the prediction task, or the information contained in attributive triples is difficult to use by the models.

8 Conclusion and Future Work

In this work, we investigated the capability of LP models that incorporate numerical literal information and the suitability of the corresponding benchmark datasets. We propose a methodology to create semi-synthetic datasets and a dataset ablation methodology.

With a semi-synthetic dataset we showed that many models underutilize literal information, even in a setting where the numerical data is crucial for the prediction. We showed that under the established evaluation schema, the performance gains of many models can be attributed to the additional model parameters rather than the models's capabilities to exploit literals.

Future work could investigate real-world KGs regarding their suitability for evaluating numerical LP more deeply. Additionally, developing more challenging synthetic dataset extensions requiring the combination of literal information and graph structure for predicting missing links could offer valuable insights into the potentially more advanced LP models proposed in the future.

9 Limitations

We see three limitations of our work:

i) Our synthetic dataset implements one simple learning goal that requires the model to learn a threshold value to make correct predictions. This learning goal is simple and does not provide any information about the models' capabilities in understanding more complex scenarios. More complex learning goals could go beyond numerical literals and could also require to combine information from numerical literals and relational triples. However, we did not investigate complex learning goals, as we believe that if models fail in simple scenarios, they will also fail in more complex ones.
ii) We did not find a model that consistently shows benefits from the numerical literals provided by the existing benchmark datasets, not even in the relational triples ablation scenario. Therefore, we can not make any conclusions about the value of the literals provided for these datasets. The numerical literals are either not relevant for the prediction task, or the information contained is difficult to use by the existing models.
iii) We exclude the evaluation of Graph Neural Network models, such as R-GCN [21], which have the ability to process numerical literals as node features. This decision is based on the absence of published research specifically advocating for these models' application in LP with numerical literal data.

Supplemental Material Statement

You can find all our source code, datasets, training result logs, and visualization Jupyter Notebooks on GitHub.[14]

Acknowledgements. This work was supported by: the Ministry of Culture and Science of the State of North Rhine-Westphalia (Germany) through SAIL, Grant No. NW21-059A; the Deutsche Forschungsgemeinschaft (DFG) through the priority program RATIO (SPP-1999), Grant No. 376059226; the Research Council of Norway through its Centres of Excellence scheme, Integreat – Norwegian Centre for knowledge-driven machine learning, project number 332645.

[14] See https://github.com/moritzblum/LiteralEvaluation.

Ethical Statement. We address concerns related to the experimental design and the significance of results of existing methods. Our objective is not to criticize the creators of the models and datasets, but rather to assist the community by providing practical guidance for future research.

In this work, efforts are made to interpret and understand how well-established models respond to changes in literal data. However, it is important to note that explainability methods still encounter challenges in interpreting such models.

All datasets utilized in this research adhere to ethical standards and are obtained from publicly available sources.

A Dataset Statistics

Table 2 shows the statistics of the evaluated LP datasets FB15k-237, YAGO3-10k, LitWD48K, and of our Synthetic dataset.

Table 2. Dataset statistics.

Dataset	**FB15k-237**	**YAGO3-10**	**LitWD48K**	**Synthetic**
# entities ($\|\mathcal{E}\|$)	14,541	123,182	47,998	14,541
# relations ($\|\mathcal{R}_E\|$)	237	37	257	237
# attributes ($\|\mathcal{R}_A\|$)	121	5	246	1
# relational triples ($\|\mathcal{G}_E\|$)	310,116	1,089,040	336,745	310,116
# attributive triples ($\|\mathcal{G}_A\|$)	70,257	111,406	148,707	14,541
# entities w/o num.	4600	31,030	8,198	0
# train	272,115	1,079,040	303,117	272,115
# test	17,535	5,000	16,838	17,535
# valid	20,466	5,000	16,838	20,466

In FB15k-237, 3/10 of the triples related to the most frequent attributive relations hold IDs for other databases, overall devoting to 6.9% of the attributive triples in the dataset. Table 3 shows the 10 most frequent attributive relations in FB15k-237.

Table 3. Ten most frequent attributive relations in FB15k-237. The original relation URIs are `http://rdf.freebase.com/ns/ + relation name`.

Relation	**# triples**
topic_server.population_number	52764
people.person.height_meters	2871
location.location.area	2166
film.film.netflix_id	1883
organization.organization.date_founded	844
user.robert.default_domain.rated_film.ew_rating	739
location.location.gnis_feature_id	645
sports.sports_team.founded	643
location.hud_county_place.countyplace_id	568
tv.tv_program.episode_running_time	493

B Hyperparameters

We either used the hyperparameters reported as best in the original publications, or performed a hyperparameter optimization in case we re-implemented a model.

We refrained from performing a new hyperparameter optimization for the models trained on our ablated datasets, as it is common practice to use the same hyperparameters across multiple datasets [13,29]. Further, Ruffinelli et al. show that the relative performance difference between various LP model architectures often shrunk through hyperparameter optimization and re-implementation when compared to prior results [18].

We use the following hyperparameters:

LiteralE$_{DistMult}$, LiteralE$_{ComplEx}$, KBLN, and MTKGNN We use the best hyperparameters reported in the original publication to ensure a fair comparison: embedding dim. 200, epochs 100, learning rate 0.001, batch size 128, embedding dropout prob. 0.2, and label smoothing 0.1. The same hyperparameters are used across all models and datasets.

TransEA. We carried out a grid-search hyperparameter optimization: embedding dim. {50, **100**}, learning rate {0.01, **0.001**}, and α {**0.1**, 0.2, ..., 0.9}.[15] The best hyperparameters are highlighted. Further, we set: epochs 500, and batch size 128.

KGA. We used the Quantile Hierarchy augmentation method, which showed the best results across all models on FB15K-237 according to [29]. For KGA$_{TuckER}$ we use the hyperparameters: embedding dim. 200, epochs 500, learning rate 0.003, batch size 128, embedding dropout prob. 0.2, and hidden dropout 0.3. For KGA$_{DistMult}$ we use additional label smoothing 0.1.

[15] We did not apply dropout or label smoothing, thereby following [31].

All models are trained for LP as usual. We applied early stopping by monitoring the MRR score on the validation set every three epochs.

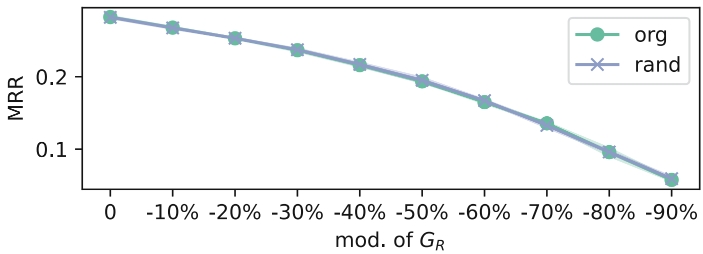
C Detailed Results

We report the MR, MRR, and Hits@10 scores (their mean and variance) obtained over three runs for models and datasets either including original features or including random features in Table 4.

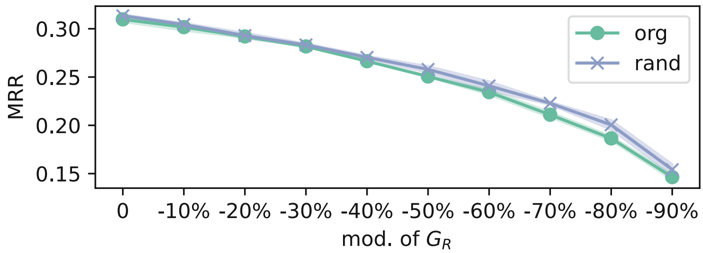
Furthermore, the effect of the relational triples ablation from FB15k-237 on MTKGNN and LiteralE$_{DistMult}$ is shown in Fig. 5.

Table 4. Comparison of scores of models trained on the datasets provided with the original literal feature versus those trained on datasets provided with random features.

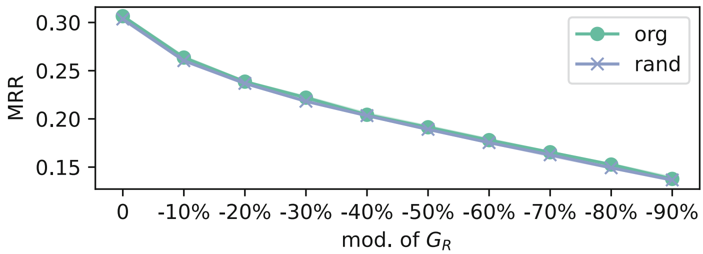
Model	**Original features**			**Random features**		
	MR	**MRR**	**Hits@1**	**MR**	**MRR**	**Hits@1**
			FB15k-237			
LiteralE$_{DistMult}$	285$\pm$001	.310$\pm$.003	.224$\pm$.003	286$\pm$008	.313$\pm$.001	.229$\pm$.002
LiteralE$_{ComplEx}$	429$\pm$002	.272$\pm$.000	.193$\pm$.001	414$\pm$022	.278$\pm$.001	.198$\pm$.001
KBLN	486$\pm$004	.295$\pm$.000	.213$\pm$.001	618$\pm$007	.285$\pm$.002	.207$\pm$.003
MTKGNN	563$\pm$006	.282$\pm$.001	.202$\pm$.001	575$\pm$004	.282$\pm$.001	.202$\pm$.001
TransEA	303$\pm$002	.203$\pm$.002	.132$\pm$.002	305$\pm$001	.200$\pm$.003	.128$\pm$.002
KGA$_{TuckER}$	200$\pm$004	.324$\pm$.001	.234$\pm$.000	200$\pm$001	.322$\pm$.001	.231$\pm$.001
KGA$_{DistMult}$	372$\pm$007	.307$\pm$.001	.221$\pm$.002	389$\pm$004	.304$\pm$.001	.220$\pm$.002
			YAGO3-10			
LiteralE$_{DistMult}$	1860$\pm$018	.479$\pm$.003	.400$\pm$.004	1925$\pm$058	.470$\pm$.003	.388$\pm$.004
LiteralE$_{ComplEx}$	2086$\pm$034	.475$\pm$.002	.400$\pm$.002	2303$\pm$116	.480$\pm$.004	.408$\pm$.004
KBLN	2850$\pm$129	.485$\pm$.023	.405$\pm$.024	4243$\pm$057	.483$\pm$.001	.401$\pm$.001
MTKGNN	3287$\pm$048	.449$\pm$.017	.362$\pm$.016	3235$\pm$094	.467$\pm$.001	.379$\pm$.002
TransEA	2226$\pm$041	.084$\pm$.002	.048$\pm$.001	2325$\pm$055	.084$\pm$.001	.049$\pm$.001
KGA$_{TuckER}$	1046$\pm$003	.497$\pm$.004	.412$\pm$.005	1094$\pm$020	.492$\pm$.002	.407$\pm$.002
KGA$_{DistMult}$	1554$\pm$005	.495$\pm$.003	.407$\pm$.004	1696$\pm$071	.496$\pm$.002	.410$\pm$.004
			LitWD48k			
LiteralE$_{DistMult}$	886$\pm$060	.326$\pm$.003	.250$\pm$.003	875$\pm$041	.330$\pm$.004	.250$\pm$.001
LiteralE$_{ComplEx}$	1489$\pm$109	.268$\pm$.005	.200$\pm$.007	1358$\pm$067	.305$\pm$.005	.238$\pm$.003
KBLN	1741$\pm$042	.329$\pm$.004	.246$\pm$.004	1836$\pm$044	.334$\pm$.002	.262$\pm$.002
MTKGNN	3085$\pm$208	.289$\pm$.002	.227$\pm$.003	2743$\pm$137	.297$\pm$.001	.235$\pm$.001
TransEA	947$\pm$019	.197$\pm$.003	.131$\pm$.003	952$\pm$032	.195$\pm$.005	.129$\pm$.004
KGA$_{TuckER}$	353$\pm$010	.403$\pm$.001	.315$\pm$.001	395$\pm$007	.376$\pm$.001	.293$\pm$.001
KGA$_{DistMult}$	504$\pm$005	.335$\pm$.001	.250$\pm$.001	1422$\pm$094	.227$\pm$.004	.169$\pm$.003



(a) MTKGNN



(b) LiteralE$_{DistMult}$



(c) KGA$_{DistMult}$

Fig. 5. MRR score after removing some percentage of relational triples from FB15k-237. The models are provided either with the original numerical features or with random features. Mean score and variance are shown across three runs.

D Further Ablation Experiments: Attributive Value Ablation

In order to provide further insights into the literal features of existing datasets, we apply an additional literal ablation method that allows to investigate whether the concrete literal value is important, or whether only the existence of such an attribute is important, or whether only the existence of an attribute can be taken into account by a model.

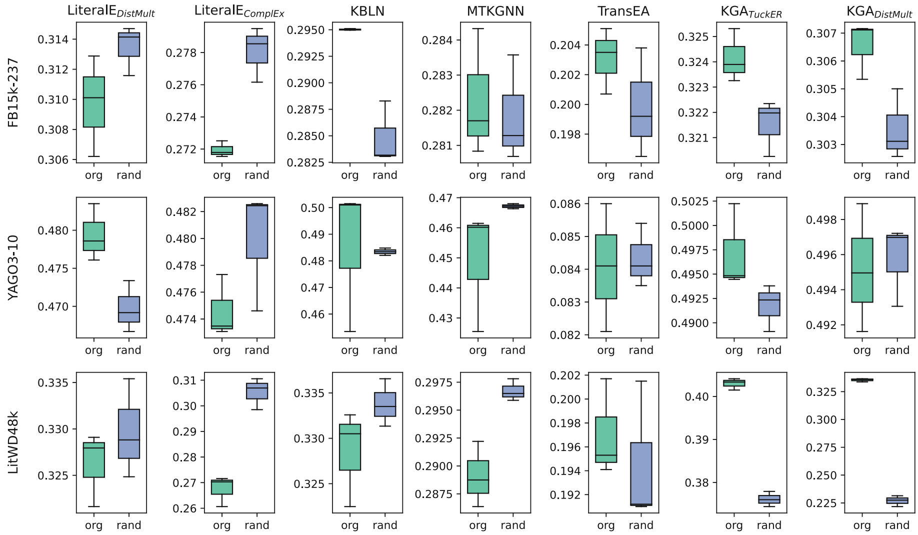


Fig. 6. MRR scores obtained over three runs for models and datasets either including original features or including random features. The variance is marginally small and not visually recognizable in the figure.

Therefore, we propose an ablation method that removes the concrete literal values but adds literal values that indicate whether the attribute exists. Given $G = G_E \cup G_A$, we create $G' = G_E \cup G'_A$ where G'_A is created as follows: if there exists a value v such that $(e, p, v) \in G_A$, then we add the triple $(e, p, 1)$ to G'_A.

We evaluate the two models, KGA$_{TuckER}$ and KGA$_{DistMult}$, which exhibit the highest benefits when provided with literals. Table 5 displays the scores of these models on three datasets. The MRR scores show no consistent trend, but the MR scores are consistently worse for the models that are provided only with the existence of attributes of entities compared to those provided with original or random features. We hypothesize that the lower performance resulting from abstracting the concrete literal values is due to the transformations that transform attributive triples into relational triples, which create a disadvantageous graph structure where most entities are connected to the two entities representing the literal values "0" and "1".

Table 5. Comparison of models trained and evaluated on datasets, each subjected to all proposed literal ablations. The model that uses the literals provided with the dataset is named *original*. The models that use only the relation type and not the concrete literal value are are named *relation type*.

	features	MR	MRR	Hits@1	Hits@3	Hits@10
FB15k-237	**KGA**$_{TuckER}$					
	original	200±.004	.324±.001	.234±.000	.355±.002	.508±.001
	random	200±.001	.322±.001	.231±.001	.354±.000	.505±.001
	relation type	217±.002	.319±.001	.229±.001	.351±.001	.502±.001
	KGA$_{DistMult}$					
	original	372±.007	.307±.001	.221±.002	.336±.001	.478±.001
	random	389±.004	.304±.001	.220±.002	.333±.001	.472±.001
	relation type	401±.011	.303±.002	.218±.002	.333±.001	.474±.005
YAGO3-10	**KGA**$_{TuckER}$					
	original	1046±.003	.497±.004	.412±.005	.546±.005	.651±.001
	random	1094±.020	.492±.002	.407±.002	.539±.002	.646±.002
	relation type	1320±.064	.521±.005	.439±.005	.570±.003	.671±.005
	KGA$_{DistMult}$					
	original	1554±.005	.495±.003	.407±.004	.543±.002	.661±.001
	random	1696±.071	.496±.002	.410±.004	.543±.002	.655±.002
	relation type	1755±.061	.509±.002	.423±.002	.559±.004	.668±.002
LitWD48k	**KGA**$_{TuckER}$					
	original	353±.010	.403±.001	.315±.001	.435±.002	.584±.000
	random	395±.007	.376±.001	.293±.001	.403±.002	.545±.002
	relation type	672±.013	.386±.001	.303±.001	.418±.000	.556±.001
	KGA$_{DistMult}$					
	original	504±.005	.335±.001	.250±.001	.360±.002	.513±.003
	random	1422±.094	.227±.004	.169±.003	.243±.005	.339±.006
	relation type	786±.015	.359±.001	.268±.001	.388±.002	.557±.003

References

1. Abboud, R., Ceylan, İ.İ., Grohe, M., Lukasiewicz, T.: The surprising power of graph neural networks with random node initialization. In: Proceedings of the 13th International Joint Conference on Artificial Intelligence (2021)
2. Balazevic, I., Allen, C., Hospedales, T.: TuckER: tensor factorization for knowledge graph completion. In: Inui, K., Jiang, J., Ng, V., Wan, X. (eds.) Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 5185–5194. Association for Computational Linguistics, Hong Kong, China (2019)
3. Blum, M., Ell, B., Cimiano, P.: Exploring the impact of literal transformations within Knowledge Graphs for Link Prediction. In: Proceedings of the 11th International Joint Conference on Knowledge Graphs, pp. 48–54. Association for Computing Machinery (2023)

4. Blum, M., Ell, B., Cimiano, P.: Insights from an OTTR-centric ontology engineering methodology. In: Proceedings of the 14th Workshop on Ontology Design and Patterns (2023)
5. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 1247–1250 (2008)
6. 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, vol. 26 (2013)
7. Dong, X., et al.: Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 601–610. Association for Computing Machinery (2014)
8. García-Durán, A., Niepert, M.: KBlrn: end-to-End Learning of Knowledge Base Representations with Latent, Relational, and Numerical Features. In: Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, pp. 372–381. AUAI Press (2018)
9. Gesese, G.A., Alam, M., Sack, H.: LiterallyWikidata - a benchmark for knowledge graph completion using literals. In: Hotho, A., et al. (eds.) ISWC 2021. LNCS, vol. 12922, pp. 511–527. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-88361-4_30
10. Gesese, G.A., Biswas, R., Alam, M., Sack, H.: A survey on knowledge graph embeddings with literals: which model links better literal-ly? Semantic Web **12**, 617–647 (2021)
11. Ismaeil, Y., Stepanova, D., Tran, T.K., Blockeel, H.: FeaBI: a Feature Selection-Based Framework for Interpreting KG Embeddings. In: Payne, T.R., et al. (eds.) ISWC 2023. LNCS, vol. 14265, pp. 599–617. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-47240-4_32
12. Khajeh Nassiri, A., Pernelle, N., Saïs, F.: REGNUM: generating logical rules with numerical predicates in knowledge graphs. In: Pesquita, C., et al. (eds.) ESWC 2023. LNCS, vol. 13870, pp. 139–155. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-33455-9_9
13. Kristiadi, A., Khan, M.A., Lukovnikov, D., Lehmann, J., Fischer, A.: Incorporating literals into knowledge graph embeddings. In: Ghidini, C., et al. (eds.) ISWC 2019. LNCS, vol. 11778, pp. 347–363. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30793-6_20
14. Mahdisoltani, F., Biega, J., Suchanek, F.: YAGO3: a knowledge base from multilingual Wikipedias. In: 7th Biennial Conference on Innovative Data Systems Research. CIDR Conference (2014)
15. Nickel, M., Tresp, V., Kriegel, H.P.: Factorizing YAGO: scalable machine learning for linked data. In: Proceedings of the 21st International Conference on World Wide Web, pp. 271–280 (2012)
16. Pan, J.Z., Razniewski, S., Kalo, J.C., et al.: Large Language Models and Knowledge Graphs: opportunities and Challenges. Trans. Graph Data Knowl. **1**, 2:1-2:38 (2023)
17. Pezeshkpour, P., Chen, L., Singh, S.: Embedding multimodal relational data for knowledge base completion. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 3208–3218. Association for Computational Linguistics (2018)

18. Ruffinelli, D., Broscheit, S., Gemulla, R.: You can teach an old dog new tricks! on training knowledge graph embeddings. In: International Conference on Learning Representations (2020)
19. Safavi, T., Koutra, D., Meij, E.: Evaluating the calibration of knowledge graph embeddings for trustworthy link prediction. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pp. 8308–8321. Association for Computational Linguistics (2020)
20. Saxena, A., Kochsiek, A., Gemulla, R.: Sequence-to-sequence knowledge graph completion and question answering. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (2022)
21. Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: Gangemi, A., et al. (eds.) ESWC 2018. LNCS, vol. 10843, pp. 593–607. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93417-4_38
22. Shirvani-Mahdavi, N., Akrami, F., Saeef, M.S., Shi, X., Li, C.: Comprehensive analysis of freebase and dataset creation for robust evaluation of knowledge graph link prediction models. In: Payne, T.R., et al. (eds.) ISWC 2023. LNCS, vol. 14266, pp. 113–133. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-47243-5_7
23. Strötgen, J., et al.: Towards the Bosch materials science knowledge base. In: Proceedings of the ISWC Satellite Tracks co-located with 18th International Semantic Web Conference, pp. 323–324. Springer (2019)
24. Tay, Y., Tuan, L.A., Phan, M.C., Hui, S.C.: Multi-task neural network for non-discrete attribute prediction in knowledge graphs. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp. 1029–1038. Association for Computing Machinery (2017)
25. Tay, Y., Tuan, L.A., Phan, M.C., Hui, S.C.: Multi-task neural network for non-discrete attribute prediction in knowledge graphs. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp. 1029-1038. Association for Computing Machinery (2017)
26. 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. Association for Computational Linguistics (2015)
27. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: International Conference on Machine Learning, pp. 2071–2080 (2016)
28. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Commun. ACM **57**, 78–85 (2014)
29. Wang, J., Ilievski, F., Szekely, P., Yao, K.T.: Augmenting knowledge graphs for better link prediction. In: Raedt, L.D. (ed.) Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22, pp. 2277–2283. International Joint Conferences on Artificial Intelligence Organization (2022). main Track
30. Wang, L., Zhao, W., Wei, Z., Liu, J.: SimKGC: simple contrastive knowledge graph completion with pre-trained language models. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, pp. 4281–4294. Association for Computational Linguistics (2022)
31. Wu, Y., Wang, Z.: Knowledge graph embedding with numeric attributes of entities. In: Proceedings of the Third Workshop on Representation Learning for NLP, pp. 132–136. Association for Computational Linguistics (2018)

32. Xue, B., Li, Y., Zou, L.: Introducing semantic information for numerical attribute prediction over knowledge graphs. In: Sattler, U., et al. (eds.) ISWC 2022. LNCS, vol. 13489, pp. 3–21. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-19433-7_1
33. Yang, B., Yih, S.W.T., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. In: Proceedings of the International Conference on Learning Representations (2015)
34. Yao, L., Mao, C., Luo, Y.: KG-BERT: BERT for knowledge graph completion. arXiv preprint arXiv:1909.03193 (2019)
35. Ying, Z., Bourgeois, D., You, J., Zitnik, M., Leskovec, J.: GNNExplainer: generating explanations for graph neural networks. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
36. Zhang, S., et al.: PaGE-link: path-based graph neural network explanation for heterogeneous link prediction. In: Proceedings of the ACM Web Conference, pp. 3784–3793 (2023)