Introducing Semantic Information for Numerical Attribute Prediction over Knowledge Graphs

Bingcong Xue[1], Yanzeng Li[1,2], and Lei Zou[1,2](✉)

[1] Peking University, Beijing, China
{xuebingcong,zoulei}@pku.edu.cn, liyanzeng@stu.pku.edu.cn
[2] Beijing Institute for General Artificial Intelligence (BIGAI), Beijing, China

Abstract. Knowledge graph (KG) completion has been long studied on link prediction task to infer missing relations, while literals are paid less attention due to the non-discrete and rich-semantic challenges. Numerical attributes such as height, age and birthday are different from other literals that they can be calculated and estimated, thus have huge potential to be predicted and play important roles in a series of tasks. However, only a few researches have made preliminary attempts to predict numerical attributes on KGs with the help of the structural information or the development of embedding techniques. In this paper, we re-examine the numerical attribute prediction task over KGs, and introduce several novel methods to explore and utilize the rich semantic knowledge of language models (LMs) for this task. An effective combination strategy is also proposed to take full advantage of both structural and semantic information. Extensive experiments are conducted to show the great effectiveness of both the semantic methods and the combination strategy.

Keywords: Numerical attribute prediction · Knowledge graph completion · Language model · Ensemble learning

1 Introduction

Knowledge graphs (KGs) store structural data typically in the form of (subject, predicate, object) triples, and have become the backbone of various AI applications such as information retrieval, question answering and recommender systems. Some well known encyclopedia KGs include DBpedia [21], Yago [29] and Wikidata [43], devoting to covering as much factual knowledge as possible. As incompleteness is inherent in all KGs and largely restricts the effectiveness, knowledge graph completion is becoming a topic of extensive research, among which link prediction is the most concerned task and knowledge graph embedding (KGE) methods play an important role.

The core idea behind KGE techniques is to map nodes and edges of KGs into a low dimensional space. The learned representation can then be used to find missing links between entities in link prediction as well as other reasoning tasks.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
U. Sattler et al. (Eds.): ISWC 2022, LNCS 13489, pp. 3–21, 2022.
https://doi.org/10.1007/978-3-031-19433-7_1

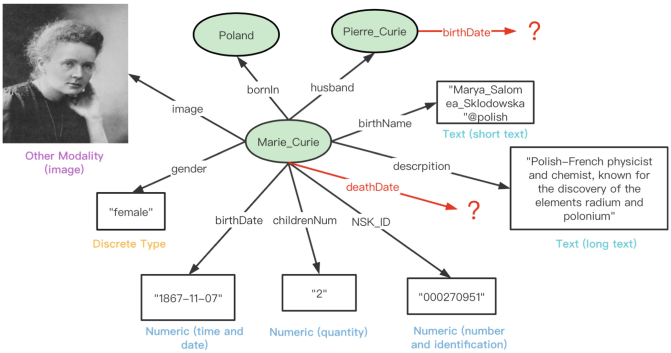

Fig. 1. A small part of a KG, where circles stand for entities and rectangles are literals. The colored text is to describe the different types of literals and the red ones are missing. (Color figure online)

According to the different mapping functions, they are roughly classified into tensor decomposition models, geometric models, and deep learning models [35]. Embedding-based methods have shown great potential in efficiently mining and analyzing on large-scale graphs, and are becoming the mainstream for knowledge graph completion task.

However, relationships among entities are not the only elements in KGs and knowledge graph completion should not be confined to just relations. For instance, various types of literal attributes also exist with rich semantics, and face the same incomplete issue. An example is depicted in Fig. 1, where an entity has not only relations with other entities, but also literal attributes in the form of text, numeric, image, etc., and all of them may be missing. In this paper, we focus on the prediction of numerical attributes over knowledge graphs, which we believe is valuable and potential but challenging as well. The motivations and intuitions are elaborated below.

1.1 Motivation

In this subsection we want to clarify our motivation by answering two questions: (1) why it is necessary to predict missing numerical attributes, and (2) why it is potential to do such a task.

1.1.1 Why do we want to predict numerical attributes? The importance of numerical attribute prediction lies in at least three aspects. Firstly, numerical attributes are widespread in KGs [39] to enrich entity characteristics from different perspectives, especially in cases of product graphs [5] and Internet of Things [12]. Like relational triplets to be completed in link prediction task, the prediction of numerical attributes itself is part of knowledge graph completion and quality management [51]. Secondly, though embedding methods have shown great potential in many reasoning tasks, traditional KGE techniques consider only relational edges and largely suffer from the sparsity problem. Introducing various literals is a powerful way to alleviate sparsity [13] and many recent researches [11,20,39,48] have shown the effectiveness to incorporate numerical attributes into the process of embedding learning. But the same incomplete problem of numerical attributes will limit the application [19]. Last but not least, numerical values can serve as the prediction targets in a chunk of standard machine learning tasks to distinguish the performance of relation representation [39], as well as language models [2,36] recently.

1.1.2 Why can numerical attributes be predicted? Different from other literals, numerical value shows the uniqueness in its ability to be compared and calculated. It is usually meaningless to approximate attribute values like an actor's name or portrait, though [32] did some attempt to decode multimodal objects with auxiliary reference inputs. But numerical attributes can be estimated even if they are not explicitly mentioned [8]. The prediction can be derived from two sources: one is the relational structure and correlation of the graph, e.g., two entities with *spouse* relation tend to have similar ages, and the other is various language models that hopefully capture and store numerical and common sense in the large-scale pre-training processes. It is our basic foothold that both the explicit structural and the implicit semantic information can produce a marked effect and experiments in Sect. 4 have demonstrated this hypothesis.

1.2 Challenges and Opportunities

Numerical attributes are much more difficult to be predicted compared with relations. Unlike the in-KG entities that are within a limited set, the values of numerical attributes are typically non-discrete, leading to the fact that if we try to encode the values into vectors for the inference, we are very likely to face a serious sparsity problem. As [39] says, the literal attributes seem to cast KGs out of its comfort zone of a bounded space. Besides, rich semantics and dependencies are hidden in the literal values that we cannot treat them as simple relational triples. And the numerical characteristics require extra calculation and comparison capabilities. If we just reduce the literals into identifiers as entity nodes, most of the information will be lost [46].

But at the same time, there are many opportunities. On the one hand, the continuous development of knowledge graph embedding techniques has shown impressive capacity for different reasoning tasks. And on the other hand, pretrained language models (PLMs) are proved to have the potential to serve as alternative knowledge bases [31,33]. And efforts on numerical reasoning in the field of natural language processing [41,52] further enhance their ability to capture and store numerical and common sense knowledge. Both the structural information behind KGs and the implicit knowledge in PLMs are promising to play a role and the integration of these two kinds of resources is in the ascendant.

1.3 Contributions

In this paper, we re-examine the less-explored numerical attribute prediction task over knowledge graphs and introduce semantic information for it. The main contributions are summarized as follows:

- We provide several novel strategies to capture the implicit knowledge behind pre-trained language models for numerical attribute prediction over KGs. To the best of our knowledge, we are the first to do such a transfer from text to graph. Compared to traditional structural methods, this line of techniques are able to capture the semantics behind literals and keep stable in zero-shot scenes, which can serve as a powerful supplement.
- After an in-depth analysis on the applicability of graph- and semantic-based methods, we design an effective combination strategy to make full use of both structural and semantic information, where base models are automatically selected for different prediction targets to achieve the best performance.
- Based on rich experimental results, we demonstrate the great effectiveness of both the semantic methods and the combination strategy. Extensive ablation studies are also conducted to show the impact of different components.

2 Preliminaries

2.1 Problem Formalization

In this subsection, we formalize the numerical attribute prediction task over KGs by first defining several key terms.

Definition 1. Knowledge Graph, denoted as $G = (E, P, L)$, is a collection of structured facts typically in the form of *(subject, predicate, object)* triples $\subseteq E \times P \times (E \bigcup L)$, where E is a set of entities, P a set of predicates and L a set of literals. A fact whose object $\in E$ is called a **relational fact**, and the corresponding predicate is called a **relation**, while a fact with a literal object is called an **attributive fact** whose predicate is known as an **attribute**.

Definition 2. Types of Literals are first presented in [13]. Like those depicted in Fig. 1, they generally fall under four kinds: (1) **text literals** of ***short text*** like names and labels, and ***long text*** such as comments and descriptions, all of which may be expressed in multiple languages; (2) **numeric literals** that are encoded as integers, float and so on, e.g., height and date; (3) **discrete types** like occupation and class, which can also be regarded as entities in some KGs, and (4) **other modalities** including images, videos and etc.

Definition 3. Numerical Attributes are a specific type of attributes whose objects are numeric literals, or in other words, numbers. They can enrich entity features in terms of quantity (like height and population), time (like birthday) and identification (like phone number and zip code).

Problem Definition. The task of numerical attribute prediction over KGs is first explored in [39] and formalized in [19]. Compared with link prediction that is to complete a missing entity for a given relation and a corresponding entity, numerical attribute prediction aims to predict the numeric value of a given entity and a given attribute. The non-discrete numerical values make it intuitively more suitable to be regarded as a regression rather than a classification problem. The task is under the context of knowledge graphs, i.e., a KG composed of a set of relational and attributive facts is given. To avoid the interference of various types of literals, the attributive facts here are limited to numerical ones. And nominal attributes [40] like the identifications are filtered out as it is typically meaningless to predict such numeric identifiers but only brings noise.

More formally, given a group of relational facts and numerical attributive facts, the task is to predict the missing numerical attribute values for a batch of entities, where the attributes are appointed and limited to non-nominal ones.

2.2 Existing Graph-Based Methods

Three preliminary jobs [1,19,39] have been done to predict numerical attributes over KGs and they are all based solely on graph structures. We summarize these graph-based methods below.

GLOBAL and **LOCAL** are two natural baselines formalized in [19]. For each type of attribute, GLOBAL predicts the missing values by the average (or median) of all the known ones, for example, all missing values of *population* will be predicted equally as the average (or median) of all the known *population* values in a given KG. And similarly, LOCAL considers the average (or median) of the same known attributes in only the neighbor nodes, and thus could get different predictions for different entities.

MRAP [1] is based on the hypothesis that a numerical attribute of entity e_a can be estimated according to e_a's other attributes as well as the attributes of e_a's surrounding entities. For instance, the *birth year* of a man seems to have some correlations with his *death year* as well as his wife's *birth year*. The correlations are modeled as regression weights iteratively estimated from the known structures, which can also be seen as a message passing scheme.

The prediction of non-discrete attributes can also be regarded as a standard regression task, where regression classifiers are trained for each attribute with some input features of the entities. The learned representations of KGE models can play a role here, and we use **KGE-reg** to stand for such a method with the learned entity embeddings serving as the features, similar to those proposed in [19,39]. The details of different KGE features are talked in Sect. 4.

3 Our Methods

3.1 Limitations of Existing Methods

Existing graph-based methods mainly depend on the interaction of the relational structures of the graph, as well as the correlations among attributes.

They ignore the semantics behind numerical values and are usually incapable of handling unseen and isolated entities. GLOBAL treats all entities equally and generally cannot obtain valuable results; LOCAL distinguishes entities based on the neighborhood structures but the simple aggregation strategy is likely to be disturbed by irrelevant noise. MRAP considers the complex interactions among various attributes and relations, which is prone to sparsity and skewness when there is a surge in the predicates number. And the message passing scheme is unfriendly to isolated entities. KGE-reg benefits from the development of various KGE methods. However, these embedding techniques are quite sensitive to the large hyper-parameter space and training strategies [4]. Though some works have published their training results, they are not always available and retraining is needed for new datasets. Also, we cannot expect to obtain good prediction results for those unseen entities during the training processes. And intuitively, not all attributes can be inferred solely from the graph structures, like the *population* of a country, which demands for some common sense and memory.

3.2 Semantic-Based Methods

We believe different types of language models, such as Bert [9], have captured and stored rich knowledge during the large-scale pre-training processes, which have been demonstrated in various natural language processing tasks. We propose semantic-based methods here to introduce the implicit semantic information of PLMs to predict missing attributes. And to better use them for our scene, we should solve two main problems: (1) how to apply them to the context of graphs, and (2) how to fully extract and utilize the implicit semantic knowledge, especially about the numerics.

Transfer for Graphs. For the first problem, we use a simple but general way to transform KG triples into meaningful texts. For a relational fact (s, p, o) in a knowledge graph, the relation predicate is converted to a natural language segment by published paraphrase dictionaries [50] or by simple heuristic rules (e.g., the predicate *happenedOnDate* is split to *happened on date*). Entities are changed from their identifications to names, and sometimes to descriptions for more semantics. Similar way runs on attributive facts, except that literal values are reserved as what they are.

Two Paradigms. For the second problem, we propose two different paradigms. The first one refers to one of the classical pre-training tasks called masked language modeling, also known as a fill-mask task. That is to say, we can change an attributive triple to be predicted into a sentence as mentioned above, leaving the missing numerical value as a masked token, which is then input to a pre-trained language model to predict a masked word. The output word is restricted to the numerical vocabulary of the model here. It actually degenerates the non-discrete numerical prediction task into a classification problem on finite digital

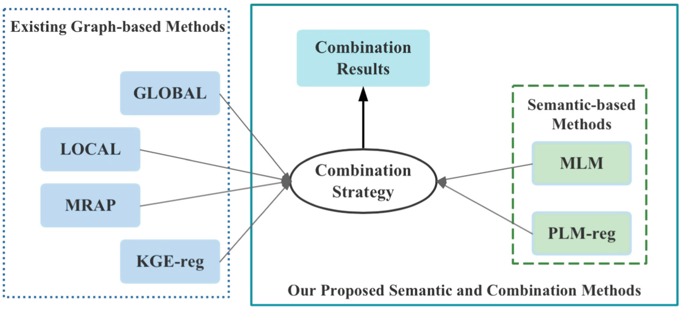

Fig. 2. Methods used in this work. The left four are based on graphs, and the right two on semantics, all of which can serve as the input to get the combination results.

tokens, and the models have no idea with the numbers, but to be tested on the implicit memory and classification abilities. And to enhance the performance of the models on specific domain tasks, fine-tuning and prompt [23] are two helpful learning techniques. The former injects domain knowledge into the model parameters, and the latter into the probe missions, which are exactly the input sentences in our task. The term **MLM** is used throughout this paper to refer to such a prediction method like cloze test, and MLM-tuning and MLM-prompt are for the two enhancement technologies respectively.

The second paradigm similar to KGE-reg, which we call **PLM-reg**, also trains attribute-specific regression classifiers for different attributes. And the difference lies in the input features, which are obtained with the help of the encoding abilities of pre-trained language models for rich semantics. We have attempted to input entity names and descriptions into PLMs and received different results, see the experimental parts for more details.

Semantic-based methods are hopeful to obtain valuable results for any input entity. And with the rapid and continuous development of language models, the ability to capture numerical semantics and predict missing values of such methods can keep growing. But all the results tend to be influenced by the paraphrasing patterns and we actually don't know exactly what the PLMs really know. Morever, MLM is limited to a fixed vocabulary and PLM-reg needs some extra resources like entity names and descriptions.

3.3 Combination Strategy

Both the graph- and semantic-based methods have some strengths and weaknesses. And a combination procedure is capable to achieve better results, where both the explicit structural and the implicit semantic knowledge are working.

As depicted in Fig. 2, we now have four graph-based methods and two semantic-based methods, which can be regarded as different base learners in the idea of ensemble learning [10]. Different models may be good at different numerical attributes, and when we put them together, the advantages of various methods can be brought into full play. We propose three combination strategies **Mean**, **Median** and **Best** respectively. In the Mean and Median strategies,

Table 1. Statistics of the datasets.

	# Ent	# Rel	# Rel_fact	# Attr	# Attr_fact	# Train	# Valid	# Test
FB15K	14,951	1,345	592,213	116	29,395	23,516	2,939	2,940
YAGO15K	15,404	32	122,886	7	23,532[a]	18,825	2,353	2,354

[a] There are 48,406 numerical facts at https://github.com/mniepert/mmkb for YAGO15K, and 23,532 are the actually left ones after removing duplicates.

Table 2. Quantities of the focused attributes following [1,19]. The upper block includes numerical attributes about time and the lower one contains all others. A dash (-) indicates that the corresponding attribute is not in the dataset.

	FB15K			YAGO15K		
	# Train	# Valid	# Test	# Train	# Valid	# Test
date_of_birth	3,528	425	475	6,555	826	837
date_of_death	988	117	115	1,490	163	169
film_release	1,479	204	184	-	-	-
organization_founded	988	126	123	-	-	-
location_founded	737	103	83	-	-	-
date_created	-	-	-	5,244	693	651
date_destroyed	-	-	-	425	55	58
date_happened	-	-	-	311	41	36
latitude	2,545	317	349	2,401	279	309
longitude	2,614	292	302	2,399	296	294
area	1,741	204	221	-	-	-
population	1,532	199	199	-	-	-
height	2,309	305	257	-	-	-
weight	182	20	24	-	-	-

the combination results are obtained as the mean and median predictions of all base models. As for the Best strategy, each attribute will choose the prediction results of the best model for it, which is measured based on the validation results. These are all model-level combination strategies, and we leave more fine-grained schemes in the future work.

4 Experiments

4.1 Experimental Setup

Datasets. We use two benchmark datasets: FB15K and YAGO15K, where the relational and numerical triples are all from MMKG [24]. We randomly divide the numerical facts into an 80/10/10 split of train/valid/test and the statistics are shown in Table 1. We follow [1,19] to focus on 11 and 7 major attributes of FB15K and YAGO15K respectively and the quantities are listed in Table 2.

Metrics. We adopt three evaluation metrics widely used in similar tasks to assess the performance: MAE (Mean Absolute Error), RMSE (Root Mean Square Error) and R^2 (R Squared), which are defined as follows:

$$MAE(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{1}$$

$$RMSE(y, \hat{y}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2}$$

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2} \tag{3}$$

where n is the sample size, y_i the ground truth of the i-th sample, $\hat{y}_i$ the predicated one and $\overline{y}$ the mean of all y values. The metrics are calculated on each type of attribute, and when evaluated on the whole, we introduce the calculation thought of micro- and macro- from the F1 metric, where the former gives the same weight to each sample and the latter to each category. MAE and RMSE reflect the deviation degree from the predictions to the true values, where smaller scores mean better. R^2 represents the proportion of variance that has been explained by the independent variables in the model and is a measure of how well unseen samples are likely to be predicted. The best possible score for R^2 is 1.0 and negative values imply the model fits much worse.

Implementation Details. As shown in Fig. 2, the methods to be compared generally fall under three headings: graph-based, semantic-based, and combination ways. For all the methods, the performances are evaluated on the test set and the validation set is used for hyper-parameters and model selection. The implementations of GLOBAL, LOCAL and MRAP methods refer to MRAP[1]. For both KGE-reg and PLM-reg, we choose three classical regression models, namely linear, ridge and lasso, from scikit-learn [28], with the complexity parameter α among [0.1, 1.0, 10.0]. We use the published KGE embeddings from LibKGE [4] and PLM models from Transformers [47], where TransE[2] and bert-base-uncased[3] are the default respectively and more other models are experimented in Sect. 4.3. The fine-tuning parameters of MLM-tuning are set by reference to [2], with a batch-size of 32 for 10 epochs and two learning rates $\{3e^{-5}, 1e^{-2}\}$, and we found empirically that more epochs would not bring further improvement. Besides, the name and description texts of FB15K entities are from DKRL [49], and the lack resources for YAGO15K are supplemented by aligning to FB15K entities according to the published sameAs links[4]. Experiments are all conducted on a

[1] https://github.com/bayrameda/MrAP.

[2] http://web.informatik.uni-mannheim.de/pi1/iclr2020-models/fb15k-237-transe.pt.

[3] https://huggingface.co/bert-base-uncased.

[4] https://github.com/nle-ml/mmkb/blob/master/YAGO15K/.

Linux machine with two NVIDIA Tesla P100 GPUs. We make all our datasets and implementations publicly available[5].

Table 3. Main results of different methods. For each dataset, the three blocks top to bottom contain graph-based, semantic-based and combination methods respectively. Best results in each block are underlined and the best ones of all methods are in boldface. Text in parentheses behind *PLM-reg* indicates the type of inputs to PLMs.

	Methods	micro-			macro-		
		MAE ↓	RMSE ↓	R^2 ↑	MAE ↓	RMSE ↓	R^2 ↑
FB15K[a]	GLOBAL	35.7281	85.5691	−0.0031	46.8625	114.6660	−0.0061
	LOCAL	21.8207	90.9444	0.3755	37.5387	138.3979	0.1270
	MRAP	17.5514	81.9242	−6.9458	30.9281	118.6432	−5.9687
	KGE-reg	28.2156	70.6051	0.4492	41.4302	99.8194	0.3773
	MLM	312.6412	698.1551	−772.3746	265.0596	625.1898	−502.4600
	MLM-tuning	32.1816	78.7322	−0.3053	35.4254	94.1896	−0.1929
	PLM-reg (name)	28.6963	73.3825	0.2947	40.9169	101.8967	0.2481
	PLM-reg (desc)	22.5595	55.8076	0.6072	33.3209	80.5485	0.5647
	Combination_Mean	19.8698	54.3243	0.3508	29.4875	78.9829	0.3188
	Combination_Median	16.0400	**51.4285**	**0.6591**	26.1637	**76.7629**	**0.5729**
	Combination_Best	**12.7935**	53.0444	0.6267	**21.3944**	78.9087	0.5717
YAGO15K	GLOBAL	49.5822	102.8896	−0.0045	49.0409	100.5088	−0.0157
	LOCAL	56.4510	123.1791	0.1312	47.9265	104.5093	0.1999
	MRAP	31.5875	86.7825	0.4539	33.1130	89.2587	0.0045
	KGE-reg	36.9135	87.7188	0.3423	37.6362	86.6269	0.3398
	MLM	187.0013	496.7505	−749.3612	217.3499	563.1464	−821.1300
	MLM-tuning	36.8188	93.1231	0.0596	34.2217	80.9421	0.1579
	PLM-reg (name)	37.9548	89.5944	0.2997	37.2637	88.0866	0.3060
	PLM-reg (desc)	32.4495	81.3838	0.4894	33.1313	80.0946	0.4755
	Combination_Mean	28.8185	**76.3485**	**0.5699**	28.4325	**68.3501**	0.6087
	Combination_Median	26.2166	79.9005	0.5445	25.2935	75.1677	0.5937
	Combination_Best	**25.2432**	82.8491	0.5218	**21.1966**	69.3299	**0.6584**

[a] Experiments show that the results of two attributes, *area* and *population*, vary largely with others. To have a better overview here, we omit these two attributes in the micro- and macro- metrics. And the detailed results can be found in Sect. 4.4.

4.2 Main Results

Table 3 reports the results of different methods for the two datasets, from which we can get the following observations. Firstly, for graph-based methods, MRAP and KGE-reg generally outperform GLOBAL and LOCAL in almost all metrics, showing the learning processes for both the interaction weights and the graph embeddings have capture valuable information for numerical attribute prediction. MRAP performs quite good on the MAE metrics, but when it comes to RMSE and R^2, it often loses to KGE-reg.

[5] https://github.com/xbc0112/NumericalPrediction.

Secondly, we can observe that, PLM-reg with entity descriptions consistently achieves the best results on both datasets and all metrics in semantic-based methods. And it also has comparable or better performances with the optimal results of graph-based methods, demonstrating the huge potential of language models for this task. The advantages will be more prominent in zero-shot scenes, since the PLMs can output stable results for any input, while other means are vulnerable to unseen or isolated entities. It is not surprising that the pure MLM performs much worse than all other methods, where it makes use of nothing but the memory of the model to classify on a limited numeric vocabulary, having no idea with the numbers as well as the input dataset. But we also find that when we just fine-tune the PLMs with the known attributes, the performances are significantly improved to be comparable with KGE-reg, which again proves that the PLMs are quite helpful and appropriate ways to extract the implicit knowledge matter much. Morever, in the implementation of PLM-reg, using descriptions brings a further performance improvement compared with the entity names, which conforms to the basic cognition that PLMs are good at capturing information from contextual texts and longer descriptions function better.

Table 4. Ablation results on KGE models for KGE-reg.

	Link Prediction			FB15K		YAGO15K	
	MRR↑	Hits@1↑	Hits@10↑	micro-MAE↓	macro-MAE↓	micro-MAE↓	macro-MAE↓
Random	-	-	-	36.3266	48.4124	49.6527	49.6824
TransE	0.313	0.221	0.497	28.2156	41.4302	36.9135	**37.6362**
RESCAL	**0.356**	**0.263**	**0.541**	28.4982	41.7494	38.7561	41.5883
ComplEx	0.348	0.253	0.536	26.4450	37.9365	38.5046	39.2547
RotatE	0.333	0.240	0.522	**25.5822**	**36.7313**	**36.3934**	37.7898

Finally, the experimental results fully reflect the great advantages of the combination methods. The combinations are conducted by excluding the three austere baselines (GLOBAL, LOCAL and MLM) and the Best selection strategy is measured on the MAE metrics. From Table 3 we can see that all of the three combination strategies greatly improve the performances on all metrics, and the Best strategy is generally the top performer, with the MAE a 20+% and a 30+% improvement on micro- and macro- metrics respectively. And similarly, if we choose the best model according to the RMSE or R^2, we could get further improvements on these metrics as well.

In general, the main results have demonstrated that the semantic-based methods are quite promising to predict numerical attributes over KGs and effective combination strategies making use of both structural and semantic knowledge can significantly improve the performances, which confirm our original motivation and the efficacy of our methods.

4.3 Ablation Study

In this subsection we conduct several ablation studies to explore the impact that the different variants of each module have on the performances, including KGE models, language models, fine-tuning parameters and description texts.

Ablation on KGE Models. Four popular KGE techniques in link prediction are chosen here for KGE-reg, namely, TransE [3], RESCAL [27], ComplEx [42] and RotatE [38]. We use the published models for FB15K from LibKGE and YAGO15K entities are mapped by the SameAs links. The official link prediction results from LibKGE as well as our KGE-reg results for two datasets are listed in Table 4, where we use Random to represent the method with random embeddings. From Table 4 we can see that the results of link prediction and numerical attribute prediction vary among different models and datasets. Though RESCAL performs best on link prediction, its performance on our task is off. At the same time, TransE and RotatE have some satisfactory results on numerical attribute predication but they are inconsistent on the two datasets. This indicates that KGE models may also lose some useful information when just focusing on certain tasks and capabilities, and numeric prediction can serve as an additional assessment, as we have talked in Sect. 1.1.1.

Table 5. Ablation results on language models for MLM.

		b-base	b-large	r-base	r-large	x-base	x-large	numBert
FB15K	micro-MAE↓	312.64	**106.87**	593.10	730.29	889.76	320.48	1,158.64
	macro-MAE↓	265.06	**101.81**	853.13	802.97	947.71	416.67	968.25
YAGO15K	micro-MAE↓	187.00	**180.26**	820.10	1,048.00	1,387.61	688.65	1,069.85
	macro-MAE↓	217.35	**134.10**	896.16	1,035.58	1,321.77	473.88	944.27

Table 6. Ablation results on fine-tuning parameters for MLM-tuning.

	FB15K		YAGO15K	
	micro-MAE↓	macro-MAE↓	micro-MAE↓	macro-MAE↓
No tuning	312.6412	265.0596	187.0013	217.3499
lr=3e−5	**32.1816**	**35.4254**	**36.8188**	**34.2217**
lr=1e−2	1,008.2838	1,081.8034	1,454.5918	1,408.1344

Table 7. Ablation results on multilingual description texts for PLM-reg. (E, F, G are English, French and German for short.)

	E	F	G	E+F	E+G	F+G	E+F+G
micro-MAE↓	22.4099	25.5759	26.0673	22.4126	**21.7161**	24.5054	22.1151
macro-MAE↓	33.5482	35.8706	37.5044	32.9257	**32.2438**	35.4397	32.6784

Ablation on Language Models. We explore the MLM results with various pre-trained language models here, including bert-base/large-uncased [9], roberta-base/large [25], xlm-roberta-base/large [7], and numBert [52]. As shown in Table 5, bert-large-uncased performs best on the two datasets, but the results are still far from satisfactory. And other carefully decorated variants of Bert even produce much worse results, which again illustrates that a pure MLM is not suitable for this task at all.

Ablation on Fine-Tuning Parameters. The impact of fine-tuning parameters (specifically learning rate here) is shown in Table 6. We can see that fine-tuning pre-trained language models with appropriate parameters will significantly improve the numerical prediction results, but on the contrary, poor configurations may bring negative effects. This reveals an inherent defect of PLM-tuning that the parameters can be difficult to choose.

Table 8. Fine-grained MAE results of five methods and the chosen model according to the Best strategy on FB15K. The numbers in bold indicate the best among all methods.

	MRAP	KGE-reg	MLM-tuning	PLM-reg(name)	PLM-reg(desc)	Best Model
date_of_birth	**13.7524**	27.0335	17.8177	28.0877	25.0356	MRAP
date_of_death	**14.1559**	67.0116	22.8152	59.8208	46.8587	MRAP
film_release	5.5087	5.0874	14.3519	11.8329	**4.9622**	PLM-reg (desc)
organization_founded	73.7679	55.7411	**39.5332**	46.5200	46.9082	PLM-reg (desc)
location_founded	152.4245	172.2755	**100.1074**	172.2287	144.9887	MLM-tuning
latitude	**2.2707**	9.7633	5.9728	8.8821	5.6201	MRAP
longitude	**4.8890**	25.1610	106.7638	29.9472	16.2546	MRAP
area	3.01e+6	2.37e+6	**5.77e+5**	1.80e+6	1.54e+6	MLM-tuning
population	1.05e+7	2.22e+7	**4.43e+6**	8.52e+6	1.57e+7	MLM-tuning
height	0.4836	0.1916	**0.1263**	0.1967	0.1881	MLM-tuning
weight	11.1000	10.6064	11.3400	10.7358	**9.0717**	PLM-reg (desc)

Ablation on Description Texts. Gesese et al. [14] have explored the benefits of multilingual descriptions for link prediction, and here we use their trilingual datasets as well as the combinations for PLM-reg (desc) on FB15K. The results are listed in Table 7, by which we can generally conclude that combining multilingual descriptions as the input for PLM-reg is promising to improve the performance but the improvement is not quite significant.

4.4 Case Study

We now start a fine-grained analysis on the performances of the methods over different attributes. The MAE results on FB15K of the five models used in the combination method are listed in Table 8, where the last column is the chosen model of the Best strategy. We can observe that the chosen model for each attribute except *organization_founded* exactly has the best performance among the methods, showing the effectiveness of the selection strategy. By looking into the bold numbers and the best models, it appears that only three methods, i.e., MRAP, MLM-tuning, and PLM-reg (desc), are actually dominant in some attributes and play a role in the combination process, where only the first one is graph-based and the others are semantic-based. This can serve as additional evidence to demonstrate the potential of the semantic methods from the fine-grained aspect.

And a more interesting finding comes when we analyze the relations associated with each attribute. We find that the attributes benefit most from the graph-based methods, such as *latitude* and *longitude*, typically have strong relations making the value derivation from the graph structures possible. A practical example is that many entities with *latitude* often have the relation *isLocatedIn* with other entities that they typically have similar *latitude* values. While other attributes, like the *height* of a person, intuitively have little to do with the graph structures, but are probably contained in the common sense knowledge behind the language models, as people's heights are actually in a small range. This observation partly explains why both structural and semantic information can play a role in the numerical attribute prediction task. And on the other hand, it inspires that we may obtain useful rules from the performance differences of the two paradigms. For instance, we may get an inference rule that *if A is located in B, then A's latitude is similar to B* here. Rule discovery is an important research problem and we will explore it further in the future.

5 Related Work

Numerical Attributes on Knowledge Graphs. Up to now, three works in total have paid attention to predicting numerical attributes over KGs. Tay et al. [39] use the learned embeddings of relational representation approaches as features to train attribute-specific regression models. It is the first to treat non-discrete numeric values as a prediction target and evaluate the performance of different models by the task of attribute value prediction. They also design a novel multi-task neural network to jointly learn from relational and numerical attribute information and experiments show that these two kinds of information are complementary to each other. The work [19] formalizes the numerical attribute prediction problem with the Global and Local baselines, and leverages knowledge graph embedding vectors in a linear regression model to get a better performance. And recently MRAP [1], a multi-relational attribute propagation algorithm in the message passing scheme, is proposed to impute missing numerical values by the learned regression model depending on the graph structure and known attributes. These works are pioneers for numerical attribute prediction over KGs and are regarded as baselines in our experiments. However, all of them focus only on the graph structures and ignore rich semantic information under numeric attributes or external resources like PLMs, and thus have a poor performance, especially in cases of unseen and isolated entities.

Another research line concerns the use of numerical attributes for representation learning [11,20,48]. For example, LiteralE [20] extends existing latent feature models with learnable parameters to incorporate numeric literals into entity embeddings, and gets performance gains in several link prediction benchmarks. These works show the utility of numerical attributes for KGE techniques, which facilitate one of our motivations to predict missing numerics.

Numerical Reasoning in Text Context. Several research topics about numerical prediction and reasoning are thriving in the field of natural language processing in recent days. One line parallel with our task is to predict missing numbers in the context of text. An early work [16] adopts Word2vec embeddings [26] of entity names as input features to regression models for number prediction. Recent empirical investigations [2,37] devote to explore the effectiveness of different combinations of various encoders and regression models. Masked numeral predication task is also used to evaluate language models' ability to capture and memorize numerical knowledge [36,52]. These methods can not be directly applied to numerical prediction over KGs and some effective ways are needed to realize the transfer, which is one of our contributions.

Some probing work has noticed the limitations of existing pre-trained language models on numerical reasoning [34,44] and then several attempts follow to inject such skills into the models by different pre-training or fine-tuning patterns, such as numBert [52], genBert [15] and numGPT [17], which can be regarded as substitutions of the basic Bert model and hopeful to further improve the performance of our method.

PLM and KG. As two major sources of knowledge playing significant roles in a series of AI applications, pre-trained language models and knowledge graphs are recently considered to be complementary to each other and can sometimes work together. On the one hand, pre-trained language models have shown potential to serve as substitute for explicit knowledge bases [31,33] or improve the performance of knowledge representation [53]. And on the other hand, some work [6,30] tries to integrate structured knowledge of KGs into current language models for better interpretability. Combining both explicit and implicit knowledge also shows advantages in tasks like recommender systems [22] and graph completion [18,45]. We are the first to explore such intergation on numerical attribute prediction and experimental results demonstrate the effectiveness of our combination strategy.

6 Conclusion and Future Work

In this paper, we focus on the prediction of numerical attributes over knowledge graphs and devote to introducing semantic information for it. Several novel semantic methods as well as effective combination strategies are proposed, and extensive experiments have shown that both the explicit structural knowledge and the implicit semantic information can help the prediction and an effective combination is of great potential.

Several interesting directions are left for the future. First, we plan to take a deep look at the paraphrase method when converting KG triples into texts, and attempt other paradigms for the use of PLMs, such as prompt. Second, fine-grained combination strategies and the value of numerical attributes on other tasks can be further explored. Last but not least, rule discovery by the compare between PLM and KG seems quite promising.

Supplemental Material Statement: Source code, datasets and results are all available at https://github.com/xbc0112/NumericalPrediction.

Acknowledgments. This work was supported by NSFC under grant 61932001, U20A20174.

References

1. Bayram, E., García-Durán, A., West, R.: Node attribute completion in knowledge graphs with multi-relational propagation. In: ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3590–3594. IEEE (2021)
2. Berg-Kirkpatrick, T., Spokoyny, D.: An empirical investigation of contextualized number prediction. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 4754–4764 (2020)
3. 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)
4. Broscheit, S., Ruffinelli, D., Kochsiek, A., Betz, P., Gemulla, R.: LibKGE - A knowledge graph embedding library for reproducible research. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 165–174 (2020). https://www.aclweb.org/anthology/2020.emnlp-demos.22
5. Cheng, K., Li, X., Xu, Y.E., Dong, X.L., Sun, Y.: Pge: Robust product graph embedding learning for error detection. arXiv preprint arXiv:2202.09747 (2022)
6. Colon-Hernandez, P., Havasi, C., Alonso, J., Huggins, M., Breazeal, C.: Combining pre-trained language models and structured knowledge. arXiv preprint arXiv:2101.12294 (2021)
7. Conneau, A., et al.: Unsupervised cross-lingual representation learning at scale. arXiv preprint arXiv:1911.02116 (2019)
8. Davidov, D., Rappoport, A.: Extraction and approximation of numerical attributes from the web. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 1308–1317 (2010)
9. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
10. Dong, X., Yu, Z., Cao, W., Shi, Y., Ma, Q.: A survey on ensemble learning. Front. Comput. Sci. **14**(2), 241–258 (2019). https://doi.org/10.1007/s11704-019-8208-z
11. García-Durán, A., Niepert, M.: KBLRN: End-to-end learning of knowledge base representations with latent, relational, and numerical features. arXiv preprint arXiv:1709.04676 (2017)

12. Gesese, G.A.: Leveraging literals for knowledge graph embeddings. In: Proceedings of the Doctoral Consortium at ISWC 2021, co-located with 20th International Semantic Web Conference (ISWC 2021), Ed.: V. Tamma. p. 9 (2021)
13. 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**(4), 617–647 (2021)
14. Gesese, G.A., Hoppe, F., Alam, M., Sack, H.: Leveraging multilingual descriptions for link prediction: Initial experiments. In: ISWC (Demos/Industry) (2020)
15. Geva, M., Gupta, A., Berant, J.: Injecting numerical reasoning skills into language models. arXiv preprint arXiv:2004.04487 (2020)
16. Gupta, A., Boleda, G., Baroni, M., Padó, S.: Distributional vectors encode referential attributes. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 12–21 (2015)
17. Jin, Z., et al.: NumGPT: Improving numeracy ability of generative pre-trained models. arXiv preprint arXiv:2109.03137 (2021)
18. Kim, B., Hong, T., Ko, Y., Seo, J.: Multi-task learning for knowledge graph completion with pre-trained language models. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 1737–1743 (2020)
19. Kotnis, B., García-Durán, A.: Learning numerical attributes in knowledge bases. In: Automated Knowledge Base Construction (AKBC) (2018)
20. Kristiadi, A., Khan, M.A., Lukovnikov, D., Lehmann, J., Fischer, A.: Incorporating literals into knowledge graph embeddings. In: Ghidni, C. (ed.) ISWC 2019. LNCS, vol. 11778, pp. 347–363. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30793-6_20
21. Lehmann, J., et al.: Dbpedia-a large-scale, multilingual knowledge base extracted from wikipedia. Semantic web **6**(2), 167–195 (2015)
22. Lian, J., Zhou, X., Zhang, F., Chen, Z., Xie, X., Sun, G.: xDeepFM: Combining explicit and implicit feature interactions for recommender systems. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, pp. 1754–1763 (2018)
23. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G.: Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. arXiv preprint arXiv:2107.13586 (2021)
24. Liu, Y., Li, H., Garcia-Duran, A., Niepert, M., Onoro-Rubio, D., Rosenblum, D.S.: MMKG: multi-modal knowledge graphs. In: Hitzler, P., Fernández, M., Janowicz, K., Zaveri, A., Gray, A.J.G., Lopez, V., Haller, A., Hammar, K. (eds.) ESWC 2019. LNCS, vol. 11503, pp. 459–474. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21348-0_30
25. Liu, Y., et al.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
26. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
27. Nickel, M., Tresp, V., Kriegel, H.P.: A three-way model for collective learning on multi-relational data. In: Icml (2011)
28. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
29. Pellissier Tanon, T., Weikum, G., Suchanek, F.: YAGO 4: a reason-able knowledge base. In: Harth, A., Kirrane, S., Ngonga Ngomo, A.-C., Paulheim, H., Rula, A., Gentile, A.L., Haase, P., Cochez, M. (eds.) ESWC 2020. LNCS, vol. 12123, pp. 583–596. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49461-2_34

30. Peters, M.E., Neumann, M., Logan IV, R.L., Schwartz, R., Joshi, V., Singh, S., Smith, N.A.: Knowledge enhanced contextual word representations. arXiv preprint arXiv:1909.04164 (2019)
31. Petroni, F., Rocktäschel, T., Lewis, P., Bakhtin, A., Wu, Y., Miller, A.H., Riedel, S.: Language models as knowledge bases? arXiv preprint arXiv:1909.01066 (2019)
32. Pezeshkpour, P., Chen, L., Singh, S.: Embedding multimodal relational data for knowledge base completion. arXiv preprint arXiv:1809.01341 (2018)
33. Roberts, A., Raffel, C., Shazeer, N.: How much knowledge can you pack into the parameters of a language model? arXiv preprint arXiv:2002.08910 (2020)
34. Rogers, A., Kovaleva, O., Rumshisky, A.: A primer in BERTology: What we know about how BERT works. Trans. Assoc. Comput. Linguist. **8**, 842–866 (2020)
35. Rossi, A., Barbosa, D., Firmani, D., Matinata, A., Merialdo, P.: Knowledge graph embedding for link prediction: a comparative analysis. ACM Trans. Knowl. Discovery Data (TKDD) **15**(2), 1–49 (2021)
36. Sakamoto, T., Aizawa, A.: Predicting numerals in natural language text using a language model considering the quantitative aspects of numerals. In: Proceedings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, pp. 140–150 (2021)
37. Spithourakis, G.P., Riedel, S.: Numeracy for language models: Evaluating and improving their ability to predict numbers. arXiv preprint arXiv:1805.08154 (2018)
38. Sun, Z., Deng, Z.H., Nie, J.Y., Tang, J.: Rotate: Knowledge graph embedding by relational rotation in complex space. arXiv preprint arXiv:1902.10197 (2019)
39. 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 (2017)
40. Thawani, A., Pujara, J., Ilievski, F.: Numeracy enhances the literacy of language models. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 6960–6967 (2021)
41. Thawani, A., Pujara, J., Szekely, P.A., Ilievski, F.: Representing numbers in nlp: a survey and a vision. arXiv preprint arXiv:2103.13136 (2021)
42. 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. PMLR (2016)
43. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Commun. ACM **57**(10), 78–85 (2014)
44. Wallace, E., Wang, Y., Li, S., Singh, S., Gardner, M.: Do nlp models know numbers? probing numeracy in embeddings. arXiv preprint arXiv:1909.07940 (2019)
45. Wang, L., Zhao, W., Wei, Z., Liu, J.: SimKGC: Simple contrastive knowledge graph completion with pre-trained language models. arXiv preprint arXiv:2203.02167 (2022)
46. Wilcke, X., Bloem, P., de Boer, V., van't Veer, R.: End-to-end learning on multimodal knowledge graphs (2021)
47. Wolf, T., et al.: Transformers: State-of-the-art natural language processing. In: Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations, pp. 38–45 (2020)
48. 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 (2018)
49. Xie, R., Liu, Z., Jia, J., Luan, H., Sun, M.: Representation learning of knowledge graphs with entity descriptions. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30 (2016)

50. Xue, B., Hu, S., Zou, L., Cheng, J.: The value of paraphrase for knowledge base predicates. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 9346–9353 (2020)
51. Xue, B., Zou, L.: Knowledge graph quality management: a comprehensive survey. In: IEEE Transactions on Knowledge and Data Engineering (2022)
52. Zhang, X., Ramachandran, D., Tenney, I., Elazar, Y., Roth, D.: Do language embeddings capture scales? arXiv preprint arXiv:2010.05345 (2020)
53. Zhang, Z., Liu, X., Zhang, Y., Su, Q., Sun, X., He, B.: Pretrain-KGE: learning knowledge representation from pretrained language models. In: Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 259–266 (2020)