Comprehensive Analysis of Freebase and Dataset Creation for Robust Evaluation of Knowledge Graph Link Prediction Models

Nasim Shirvani-Mahdavi, Farahnaz Akrami, Mohammed Samiul Saeef, Xiao Shi, and Chengkai Li(✉)

University of Texas at Arlington, Arlington, TX 76019, USA
{nasim.shirvanimahdavi2,farahnaz.akrami,mohammedsamiul.saeef, xiao.shi}@mavs.uta.edu, cli@uta.edu

Abstract. Freebase is amongst the largest public cross-domain knowledge graphs. It possesses three main data modeling idiosyncrasies. It has a strong type system; its properties are purposefully represented in reverse pairs; and it uses mediator objects to represent multiary relationships. These design choices are important in modeling the real-world. But they also pose nontrivial challenges in research of embedding models for knowledge graph completion, especially when models are developed and evaluated agnostically of these idiosyncrasies. This paper lays out a comprehensive analysis of the challenges associated with the idiosyncrasies of Freebase and measures their impact on knowledge graph link prediction. The results fill an important gap in our understanding of embedding models for link prediction as such models were never evaluated using a proper full-scale Freebase dataset. The paper also makes available several variants of the Freebase dataset by inclusion and exclusion of the data modeling idiosyncrasies. It fills an important gap in dataset availability too as this is the first-ever publicly available full-scale Freebase dataset that has gone through proper preparation.

Keywords: Knowledge graph completion · Link prediction · Knowledge graph embedding · Benchmark dataset

1 Introduction

Knowledge graphs (KGs) encode semantic, factual information as triples of the form (subject `s`, predicate p, object `o`). They can link together heterogeneous data across different domains for purposes greater than what they support separately. KGs have become an essential asset to a wide variety of tasks and applications in the fields of artificial intelligence and machine learning [13,24], including natural language processing [47], search [46], question answering [22], and recommender systems [49]. Thus, KGs are of great importance to many technology companies [17,30] and governments [3,29].

T. R. Payne et al. (Eds.): ISWC 2023, LNCS 14266, pp. 113–133, 2023.
https://doi.org/10.1007/978-3-031-47243-5_7

To develop and robustly evaluate models and algorithms for tasks on KGs, access to large-scale KGs is crucial. But publicly available KG datasets are often much smaller than what real-world scenarios render and require [23]. For example, FB15k and FB15k-237 [10,39], two staple datasets for knowledge graph completion, only have less than 15,000 entities in each. As of now, only a few cross-domain common fact KGs are both large and publicly available, including DBpedia [7], Freebase [8], Wikidata [41], YAGO [37], and NELL [12].

With more than 80 million nodes, Freebase is amongst the largest public KGs. It comprises factual information in a broad range of domains. The dataset possesses several data modeling idiosyncrasies which serve important practical purposes in modeling the real-world. *Firstly*, Freebase properties are purposefully represented in reverse pairs, making it convenient to traverse and query the graph in both directions [31]. *Secondly*, Freebase uses mediator objects to facilitate representation of n-ary relationships [31]. *Lastly*, Freebase's strong de facto type system categorizes each entity into one or more types, the type of an entity determines the properties it may possess [9], and the label of a property *almost* functionally determines the types of the entities at its two ends.

Albeit highly useful, the aforementioned idiosyncrasies also pose nontrivial challenges in the advancement of KG-oriented technologies. Specifically, when algorithms and models for intelligent tasks are developed and evaluated agnostically of these data modeling idiosyncrasies, one could either miss the opportunity to leverage such features or fall into pitfalls without knowing. One example is that for knowledge graph link prediction—the task of predicting missing `s` in triple (`?`, p, `o`) or missing `o` in (`s`, p, `?`)—many models [33,42] proposed in the past decade were evaluated using FB15k, a small subset of Freebase full of reverse triple pairs. The reverse triples lead to data leakage in model evaluation. The consequence is substantial over-estimation of the models' accuracy and thus faulty and even reversed comparison of their relative strengths [5].

This paper provides four variants of the Freebase dataset by inclusion/exclusion of mediator objects and reverse triples. It also provides a Freebase type system which is extracted to supplement the variants. It lays out a comprehensive analysis of the challenges associated with the aforementioned idiosyncrasies of Freebase. Using the datasets and the type system, it further measures these challenges' impact on embedding models (e.g., TransE [10] and ComplEx [40]) which are most extensively employed for knowledge graph link prediction. Furthermore, the datasets underwent thorough cleaning in order to improve their utility and to remove irrelevant triples from the original Freebase data dump [18]. The methodology, code, datasets, and experiment results produced from this work, available at https://github.com/idirlab/freebases, are significant contributions to the research community, as follows.

The paper fills an important gap in dataset availability. To the best of our knowledge, ours is the first-ever publicly available full-scale Freebase dataset that has gone through proper preparation. Specifically, our Freebase variants were prepared in recognition of the aforementioned data modeling idiosyncrasies, as well as via thorough data cleaning. On the contrary, the Freebase data dump has

all types of triples tangled together, including even data about the operation of Freebase itself which are not common knowledge facts; Freebase86m [51], the only other public full-scale Freebase dataset, also mixes together metadata (such as data related to Freebase type system), administrative data, reverse triples, and mediator objects.

The paper also fills an important gap in our understanding of embedding models for knowledge graph link prediction. Such models were seldom evaluated using the full-scale Freebase. When they were, the datasets used (e.g., the aforementioned Freebase86m) were problematic, leading to unreliable results. The experiments on our datasets inform the research community several important results that were never known before, including 1) the true performance of link prediction embedding models on the complete Freebase, 2) how data idiosyncrasies such as mediator objects and reverse triples impact model performance on the complete Freebase data, and 3) similarly, how the mixture of knowledge facts, metadata and administrative data impact model performance.

The datasets and results are highly relevant to researchers and practitioners, as Freebase remains the single most commonly used dataset for link prediction, by far. Upon examining all full-length publications appeared in 12 top conferences in 2022, we found 53 publications used datasets commonly utilized for link prediction. The conferences, the papers, and the datasets used in them are listed in a file "papers.xlsx" in GitHub repository https://github.com/idirlab/freebases. Amongst these publications, 48 utilized datasets derived from Freebase, only 3 publications used a Freebase dataset at its full scale, specifically Freebase86m, while 8 made use of datasets from Wikidata. The properly processed full-scale Freebase datasets from this work can facilitate researchers and practitioners in carrying out large-scale studies on knowledge graph completion and beyond.

The dataset creation was nontrivial. It required extensive inspection and complex processing of the massive Freebase data dump, for which documents are scarce. None of the idiosyncrasies, as articulated in Sects. 3 and 4, was defined or detailed in the data dump itself. Figuring out the details required iterative trial-and-error in examining the data. To the best of our knowledge, more detailed description of these idiosyncrasies is not available anywhere else. If one must learn to examine Freebase and prepare datasets from scratch, the process has a steep learning curve and can easily require many months. Our datasets can thus accelerate the work of many researchers and practitioners.

The datasets and experimentation design can enable comparison with non-conventional models and on other datasets. Our methodology of processing and analyzing data is extensible to other datasets with similar data modeling idiosyncrasies, such as YAGO3-10 and WN18 which have redundant and reverse relations [5] and Wikidata which represents multiary relationships using *statements*. The experiment design could be extended to studying the impact of multiary relationships in Wikidata on various kinds of link prediction models. Further, given the datasets and experiment results made available in this paper, it becomes possible to compare the real performance of conventional embedding

models and hyper-relational fact models [20,34,44,50] on a full-scale Freebase dataset that includes multiary relationships (i.e., mediator objects).

2 Freebase Basic Concepts

This section provides a summary of basic terminology and concepts related to Freebase. We aim to adhere to [9,19,25,31] in nomenclature and notation.

RDF: Freebase is available from its data dumps [18] in N-Triples RDF (Resource Description Format) [25]. An RDF graph is a collection of triples (`s`, *p*, `o`), each comprising a *subject* `s`, an *object* `o`, and a *predicate* *p*. An example triple is (`James Ivory`, */film/director/film*, `A Room with a View`).

Topic (entity, node): In viewing Freebase as a graph, its nodes can be divided into *topics* and *non-topics*. Topics are distinct entities, e.g., `James Ivory` in Fig. 1. An example of non-topic nodes is CVT (Compound Value Type) nodes which are used to represent *n*-ary relations (details in Sect. 3). Other non-topic nodes are related to property, domain and type (see below). Each topic and non-topic node has a unique *machine identifier* (MID), which consists of a prefix (either `/m/` for Freebase Identifiers or `/g/` for Google Knowledge Graph Identifiers) followed by a base-32 identifier. For example, the MID of `James Ivory` is `/m/04ld94`. For better readability, we use the names (i.e., labels) of topics and non-topics in presenting triples in this paper. Inside the dataset, though, they are represented by MIDs.

Type and **domain:** Freebase topics are grouped into *types* semantically. A topic may have multiple types, e.g., `James Ivory`'s types include */people/person* and */film/director*. Types are further grouped into *domains*. For instance, domain *film* includes types such as */film/actor*, */film/director*, and */film/editor*.

Property (predicate, relation, edge): *Properties* are used in Freebase to provide facts about topics. A property of a topic defines a relationship between the topic and its property value. The property value could be a literal or another topic. Property labels are structured as /[domain]/[type]/[label]. The /[domain]/[type] prefix identifies the topic's type that a property belongs to, while [label] provides an intuitive meaning of the property. For example, topic `James Ivory` has the property */people/person/date_of_birth* with value 1928-06-07. This property is pertinent to the topic's type */people/person*. The topic also has another property */film/director/film*, on which the value is another topic `A Room with a View`, as shown in Fig. 1. This property is pertinent to another type of the topic—*/film/director*. A relationship is represented as a triple, where the triple's predicate is a property of the topic in the triple's subject. In viewing Freebase as a graph, a property is a directed edge from the subject node to the object node. The type of an edge (i.e., *edge type*) can be distinctly identified by the label of the edge (i.e., the property label). The occurrences of an edge type in the graph are *edge instances*.

Schema: The term schema refers to the way Freebase is structured. It is expressed through types and properties. The schema of a type is the collection of its properties. Given a topic belonging to a type, the properties in that type's schema are applicable to the topic. For example, the schema of type */people/person* includes property */people/person/date_of_birth*. Hence, each topic of this type (e.g., `James Ivory`) may have the property.

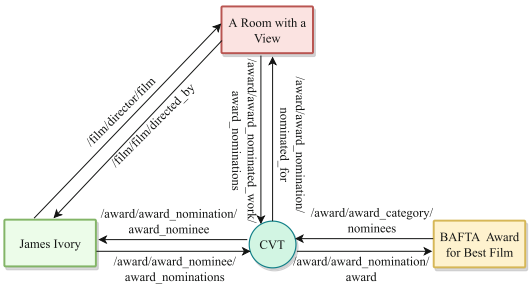


Fig. 1. A small fragment of Freebase, with a mediator node

3 Idiosyncrasies of Freebase and Challenges They Pose

Freebase is the single most commonly used dataset for the task of link prediction, as mentioned in Sect. 1. The Freebase raw data dump contains more than 80 million nodes, more than 14,000 distinct relations, and 1.9 billion triples. It has a total of 105 domains, 89 of which are diverse *subject matter domains*—domains describing real-world facts [13]. This section explains several idiosyncrasies of Freebase's data modeling design choices, and their impacts on link prediction.

3.1 Reverse Triples

When a new fact was included into Freebase, it would be added as a pair of reverse triples (`s`, p, `o`) and (`s`, p^{-1}, `o`) where p^{-1} is the reverse of p. Freebase denotes reverse relations explicitly using a special relation */type/property/reverse_property* [16,31]. For instance, */film/film/directed_by* and */film/director/film* are reverse relations, as denoted by a triple (`/film/film/directed_by`, */type/property/reverse_property*, `/film/director/film`). Thus, (`James Ivory`, */film/director/film*, `A Room With A View`) and (`A Room With A View`, */film/film/directed_by*, `James Ivory`) form reverse triples, shown as two edges in reverse directions in Fig. 1.

Several previous studies discussed the pitfalls in including reverse relations in datasets used for knowledge graph link prediction task [4,5,15,39]. The popular benchmark dataset FB15k (a relatively small subset of Freebase), created by Bordes et al. [10], was almost always used for this task. Toutanova and

Chen [39] noted that FB15k contains many reverse triples. They constructed another dataset, FB15k-237, by only keeping one relation out of any pair of reverse relations. The pitfalls associated with reverse triples in datasets such as FB15k can be summarized as follows. 1) Link prediction becomes much easier on a triple if its reverse triple is available. Hence, the reverse triples led to substantial over-estimation of model accuracy, which is verified by experiments in [5]. 2) Instead of complex models, one may achieve similar results by using statistics of the triples to derive simple rules of the form $(\mathtt{s}, p_1, \mathtt{o}) \Rightarrow (\mathtt{o}, p_2, \mathtt{s})$ where p_1 and p_2 are reverse. Such rules are highly effective given the prevalence of reverse relations [5,15]. 3) The link prediction scenario for such data is non-existent in the real-world at all. For such intrinsically reverse relations that always come in pair, there is not a scenario in which one needs to predict a triple while its reverse is already in the knowledge graph. More precisely, this is a case of excessive *data leakage*—the model is trained using features that otherwise would not be available when the model needs to be applied for real inference.

For all reasons mentioned above, there is no benefit to include reverse triples in building link prediction models. If one still chooses to include them, care must be taken to avoid the aforementioned pitfalls. Particularly, a pair of reverse triples should always be placed together in either training or test set.

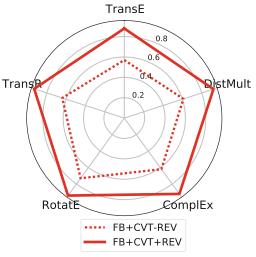


Fig. 2. Impact of reverse triples on MRR$^{\uparrow}$ of embedding models

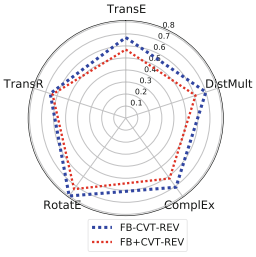


Fig. 3. Impact of mediator nodes on MRR$^{\uparrow}$ of embedding models

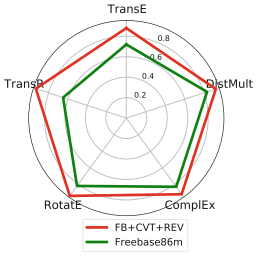


Fig. 4. Impact of non-subject matter triples on MRR$^{\uparrow}$ of embedding models

The impact of reverse triples was previously only examined on small-scale datasets FB15k and FB15k-237 [4,5,15,39]. Our corresponding experiment results on full-scale Freebase thus answer an important question for the first time. While the full results and experiment setup are detailed in Sect. 7 (specifically Table 7 and Fig. 5), here we summarize the most important observations. Figure 2 compares the performance of several representative link predication models on a commonly used performance measure MRR$^{\uparrow}$, using two new full-scale Freebase datasets created by us (details of dataset creation in Sect. 6). FB+CVT+REV is obtained after cleaning the Freebase data dump and removing irrelevant data, and in FB+CVT-REV reverse relations are further removed by only keeping one relation out of each reverse pair. Similar to the comparison

Table 1. Link prediction performance (MRR$^{\uparrow}$) on FB+CVT-REV vs. FB+CVT+REV

	FB+CVT-REV			FB+CVT+REV		
Model	unidir	bidir	all	unidir	bidir	all
TransE	0.72	0.56	0.57	0.75	0.89	0.88
DistMult	0.65	0.60	0.61	0.70	0.94	0.92
ComplEx	0.67	0.61	0.62	0.69	0.94	0.92
TransR	0.66	0.63	0.64	0.75	0.94	0.93
RotatE	0.67	0.74	0.73	0.73	0.96	0.94

Table 2. Link prediction performance (MRR$^{\uparrow}$) on FB-CVT-REV vs. FB+CVT-REV

	FB-CVT-REV			FB+CVT-REV		
Model	binary	concatenated	all	binary	multiary	all
TransE	0.60	0.90	0.67	0.57	0.96	0.57
DistMult	0.64	0.89	0.70	0.61	0.77	0.61
ComplEx	0.66	0.90	0.71	0.62	0.80	0.62
TransR	0.58	0.92	0.66	0.63	0.87	0.64
RotatE	0.76	0.92	0.80	0.73	0.88	0.73

results on small-scale FB15k vs. FB15k-237, the results on the full-scale datasets also show drastic decrease of model accuracy after removal of reverse triples and thus overestimation of model performance due to reverse triples.

We further break down the results by categorizing all relations into two groups—*unidirectional relations* (denoted as "unidir" in Table 1 and Table 3) which do not have reverse relations and *bidirectional relations* (denoted "bidir") which have reverse relations in the original Freebase data dump. In Table 1, the columns labeled "all" correspond to Fig. 2 and are for both categories of relations together. As the table shows, while the performance degradation is universal, the drop is significantly more severe for bidirectional relations due to removing reserve triples.

Table 3. Link prediction results on FB15k-237 vs. FB15k

	FB15k-237						
Model	MRR$^{\uparrow}$ (unidir)	MRR$^{\uparrow}$ (bidir)	MRR$^{\uparrow}$ (all)	MR$^{\downarrow}$	Hits@1$^{\uparrow}$	Hits@3$^{\uparrow}$	Hits@10$^{\uparrow}$
TransE	0.35	0.22	0.24	257.75	0.14	0.28	0.44
DistMult	0.31	0.23	0.24	385.12	0.14	0.27	0.43
ComplEx	0.30	0.22	0.23	425.38	0.14	0.25	0.42
TransR	0.54	0.58	0.57	196.99	0.52	0.59	0.67
RotatE	0.39	0.22	0.24	288.43	0.16	0.26	0.42
	FB15k						
Model	MRR$^{\uparrow}$(unidir)	MRR$^{\uparrow}$(bidir)	MRR$^{\uparrow}$(all)	MR$^{\downarrow}$	Hits@1$^{\uparrow}$	Hits@3$^{\uparrow}$	Hits@10$^{\uparrow}$
TransE	0.56	0.63	0.63	46.55	0.49	0.73	0.83
DistMult	0.60	0.69	0.68	59.92	0.57	0.76	0.86
ComplEx	0.59	0.76	0.74	66.37	0.66	0.81	0.88
TransR	0.63	0.66	0.66	66.09	0.57	0.72	0.80
RotatE	0.63	0.68	0.68	50.28	0.57	0.75	0.85

To put the discussion in context, we reproduced the results on FB15k and FB15k-237 using DGL-KE [51], which is the framework we used in this study for experiments on large-scale datasets. The results (Table 3) are mostly consistent with previously reported results using frameworks for small-scale datasets (e.g., LibKGE [11]), barring differences that can be attributed to implementations of different frameworks. Comparing Table 1 and Table 3, we can observe

that models' performance on full-scale datasets is significantly higher than the small-scale counterpart, unsurprisingly given the much larger datasets. What are common for both small-scale and large-scale datasets are the performance degradation due to removal of reverse triple as well as the observations regarding unidirectional vs. bidirectional relations.

3.2 Mediator Nodes

Mediator nodes, also called CVT nodes, are used in Freebase to represent n-ary relationships [31]. For example, Fig. 1 shows a CVT node connected to an `award`, a `nominee`, and a `work`. This or similar approach is necessary for accurate modeling of the real-world. Note that, one may convert an n-ary relationship centered at a CVT node into $\binom{n}{2}$ binary relationships between every pair of entities, by concatenating the edges that connect the entities through the CVT node. While such a transformation may help reduce the complexity of algorithmic solutions, it results in loss of information [44] and is irreversible [33], and thus it may not always be an acceptable approach as far as data semantics is concerned. Nevertheless, most prior studies of knowledge graph link prediction use Freebase datasets without CVT nodes, e.g., FB15k and FB15k-237, which applied the aforementioned transformation. Though lossful for Freebase-like KGs, the insights gained using such datasets could be more applicable toward graphs with only binary relationships.

When multiary relationships (i.e., CVT nodes) are present, link prediction could become more challenging as CVT nodes are long-tail nodes with limited connectivity. Nonetheless, impact of CVT nodes on the effectiveness of current link prediction approaches is unknown. This paper for the first time presents experiment results in this regard, on full-scale Freebase datasets. While Sect. 7 presents the full results, here we highlight the most important observations.

Figure 3 shows the performance ($\texttt{MRR}^{\uparrow}$) of various models on two of our new datasets, FB-CVT-REV and FB+CVT-REV (dataset details in Sect. 6). In both datasets, reverse relations are removed by keeping only one relation out of every reverse pair so that we can solely focus on the impact of CVT nodes. CVT nodes are kept in FB+CVT-REV but removed from FB-CVT-REV by the concatenation approach discussed in Sect. 6. All models performed worse when CVT nodes are present, verifying our earlier analysis.

We further broke down the results by categorizing all relations into two groups—binary relations and multiary (or concatenated) relations. Binary relations are between two regular entities. While multiary relations in FB+CVT-REV connect regular entities with CVT nodes, concatenated relations in FB-CVT-REV are the binary relations converted from multiary relations. In Table 2, the columns labeled "all" correspond to Fig. 3 and are for both categories of relations together. These results show that most models perform better on concatenated relations than multiary relations, further verifying the aforementioned challenges posed by CVT nodes. Furthermore, for all models and datasets, the models' accuracy on concatenated/multiary relations are substantially higher

than that on binary relations. This could be due to different natures of binary and multiary relations in the datasets and is worth further examination.

3.3 Metadata and Administrative Data

As stated in [13], Freebase domains can be divided into 3 groups: implementation domains, Web Ontology Language (OWL) domains, and subject matter domains. Freebase implementation domains such as */dataworld/* and */freebase/* include triples that convey schema and technical information used in creation of Freebase. According to [18], */dataworld/* is "a domain for schema that deals with operational or infrastructural information" and */freebase/* is "a domain to administer the Freebase application." For example, */freebase/mass_data_operation* in the */freebase/* domain is a type for tracking large-scale data tasks carried out by Freebase data team. OWL domains contain properties such as *rdfs:domain* and *rdfs:range* for some predicates *p*. *rdfs:domain* denotes to which type the subject of any triple of predicate *p* belongs, and *rdfs:range* denotes the type of the object of any such triple [6]. For example, the domain and range of the predicate */film/director/film* are *director* and *film*, respectively.

Different from implementation domains and OWL domains, subject matter domains contain triples about knowledge facts. We call (`s`, *p*, `o`) a subject matter triple if `s`, *p* and `o` belong to subject matter domains. Computational tasks and applications thus need to be applied on this category of domains instead of the other two categories. However, about 31% of the Freebase86m [51] triples fall under non-subject matter domains, more specifically implementation domains since OWL domains were removed from Freebase86m. These domains are listed in Table 4, to show concretely what they are about. The purposes of some of these domains were explained earlier in this section. We have created 4 datasets in which only the triples belonging to subject matter domains are retained. We also provide the information related to type system as discussed in Sect. 3. The details of this process are discussed in Sect. 6.

Table 4. Statistics of implementation domains in Freebase86m

Domain	#Triples	%Total
/common/	48,610,556	14.4
/type/	26,541,747	7.8
/base/	14,253,028	4.2
/freebase/	7,705,605	2.3
/dataworld/	6,956,819	2.1
/user/	322,215	0.1
/pipeline/	455,377	0.1
/kp_lw/	1,034	0.0003

Table 5. Link prediction performance (`MRR`$^{\uparrow}$) on Freebase86m and FB+CVT+REV

	Freebase86m			FB+CVT+REV
Model	subj matter	non-subj matter	all	all
TransE	0.74	0.68	0.72	0.88
DistMult	0.91	0.64	0.83	0.92
ComplEx	0.91	0.64	0.83	0.92
TransR	0.76	0.39	0.65	0.93
RotatE	0.92	0.56	0.82	0.94

Figure 4 shows the impact of non-subject matter triples by comparing the performance (MRR$^{\uparrow}$) of link prediction models on Freebase86m and our new dataset FB+CVT+REV, which includes only subject matter triples. The figure shows the adverse effect of non-subject matter triples. Table 5 further breaks down the results separately on subject matter and non-subject matter triples. The results clearly show that the models struggled on non-subject matter triples.

4 Freebase Type System

Freebase categorizes each topic into one or more types and each type into one domain. Furthermore, the triple instances satisfy *pseudo* constraints as if they are governed by a rigorous type system. Specifically, 1) given a node, its types set up constraints on the labels of its properties; the /[domain]/[type] segment in the label of an edge in most cases is one of the subject node's types. To be more precise, this is a constraint satisfied by 98.98% of the nodes—we found 610,007 out of 59,896,902 nodes in Freebase (after cleaning the data dump; more to be explained later in Sect. 6) having at least one property belonging to a type that is not among the node's types. 2) Given an edge type and its edge instances, there is *almost* a function that maps from the edge type to a type that all subjects in the edge instances belong to, and similarly *almost* such a function for objects. For instance, all subjects of edge *comedy/comedian/genres* belong to type */comedy/comedian* and all their objects belong to */comedy/comedy_genre*. Particularly, regarding objects, the Freebase designers explained that every property has an "expected type" [9]. For each edge type, we identified the most common entity type among all subjects and all objects in its instances, respectively. To this end, we filtered out the relations without edge labels in Freebase data dump, since the type of a property is known by its label. Given 2,891 such edge types with labels out of 3,055 relations in our dataset FB-CVT-REV (explained in Sect. 6), for 2,011, 2,510, 2,685, and 2,723 edge types, the most common entity type among subjects covers 100%, 99%, 95%, and 90% of the edge instances, respectively. With regard to objects, the numbers are 2,164, 2,559, 2,763, and 2,821, for 100%, 99%, 95%, and 90%, respectively.

Given the *almost* true constraints reflected by the aforementioned statistics, we created an explicit type system, which can become useful when enforced in various tasks such as link prediction. Note that Freebase itself does not explicitly specify such a type system, even though its data appear to follow guidelines that approximately form the type system, e.g., the "expected type" mentioned earlier. Our goal in creating the type system is to, given an edge type, designate a *required type* for its subjects (and objects, respectively) from a pool of candidates formed by all types that the subjects (objects, respectively) belong to. As an example, consider edge type */film/film/performance* and the entities o at the object end of its instances. These entities belong to types {*/film/actor*, */tv/tv_actor*, */music/artist*, */award/award_winner*, */people/person*}, which thus form the candidate pool. We select the required type for its object end in two steps, and the same procedure is applied for the subject/object ends of all edge types.

In *step 1*, we exclude a candidate type t if $P(o \in t) < \alpha$, i.e., the probability of the object end of */film/film/performance* belonging to t is less than a threshold α. The rationale is to keep only those candidates with sufficient coverage. In the dataset, $P(o \in$ */film/actor*$) = 0.9969$, $P(o \in$ */tv/tv_actor*$) = 0.1052$, $P(o \in$ */music/artist*$) = 0.0477$, $P(o \in$ */award/award_winner*$) = 0.0373$, and $P(o \in$ */people/person*$) = 0.998$. Using threshold $\alpha = 0.95$, */tv/tv_actor*, */music/artist* and */award/award_winner* were excluded. In *step 2*, we choose the most *specific* type among the remaining candidates. The most specific type is given by $\arg\min_t \sum_{t' \neq t} P(o \in t | o \in t')$, where t and t' are from remaining candidates. $P(o \in t | o \in t')$ is the conditional probability of a Freebase entity o belonging to type t given that it also belongs to type t'. In the dataset, $P(o \in$ */people/person* $\mid o \in$ */film/actor*$) = 0.9984$ and $P(o \in$ */film/actor* $\mid o \in$ */people/person*$) = 0.1394$. Thus, we assigned */film/actor* as the required entity type for objects of edge type */film/film/performance* because it is more specific than */people/person*, even though */people/person* had slightly higher coverage.

The type system we created can be useful in improving link prediction. A few studies in fact employed type information for such a goal [21,45]. Particularly, embedding models can aim to keep entities of the same type close to each other in the embedding space [21]. Further, type information could be a simple, effective model feature. For instance, given the task of predicting the objects in (`James Ivory`, */film/director/film*, `?`), knowing the object end type of */film/director/film* is */film/film* can help exclude many candidates. Finally, type information can be used as a constraint for generating more useful negative training or test examples. For instance, a negative example (`James Ivory`, */film/director/film*, `BAFTA Award for Best Film`) has less value in gauging a model's accuracy since it is a trivial case, as `BAFTA Award for Best Film` is not of type */film/film*.

5 Defects of Existing Freebase Datasets

Over the past decade, several datasets were created from Freebase. This section reviews some of these datasets and briefly discusses flaws associated with them.

FB15k [10] includes entities with at least 100 appearances in Freebase that were also available in Wikipedia based on the *wiki-links* database [14]. Each included relation has at least 100 instances. 14,951 entities and 1,345 relations satisfy these criteria, which account for 592,213 triples included into FB15k. These triples were randomly split into training, validation and test sets. This dataset suffers from data redundancy in the forms of reverse triples, duplicate and reverse-duplicate relations. Refer to [5] for a detailed discussion of such.

FB15k-237 [39], with 14,541 entities, 237 relations and 309,696 triples, was created from FB15k in order to mitigate the aforementioned data redundancy. Only the most frequent 401 relations from FB15k are kept. Near-duplicate and reverse-duplicate relations were detected, and only one relation from each pair of such redundant relations is kept. This process further decreased the number of relations to 237. This step could incorrectly remove useful information, in two scenarios. 1) False positives. For example, hypothetically *place_of_birth* and

place_of_death may have many overlapping subject-object pairs, but they are not semantically redundant. 2) False negatives. The creation of FB15k-237 did not resort to the accurate reverse relation information encoded by *reverse_property* in Freebase. For example, */education/educational_institution/campuses* and */education/educational_institution_campus/educational_institution* are both in FB15k-237 but they are reverse relations according to *reverse_property*.

Freebase86m is created from the last Freebase data dump and is employed in evaluating large-scale knowledge graph embedding frameworks [28,51]. It includes 86,054,151 entities, 14,824 relations and 338,586,276 triples. No information is available on how this dataset was created. We carried out an extensive investigation to assess its quality. We found that 1) 31% of the triples in this dataset are non-subject matter triples from Freebase implementation domains such as */common/* and */type/*, 2) 23% of the dataset's nodes are mediator nodes, and 3) it also has abundant data redundancy since 38% of its triples form reverse triples. As discussed in Sect. 3, non-subject matter triples should be removed; reverse triples, when not properly handled, lead to substantial over-estimation of link predication models' accuracy; and the existence of mediator nodes presents extra challenges to models. Mixing these different types of triples together, without clear annotation and separation, leads to foreseeably unreliable models and results. Section 7 discusses in detail the impact of these defects in Freebase86m.

6 Data Preparation

Variants of the Freebase Dataset. We created four variants of the Freebase dataset by inclusion/exclusion of reverse triples and CVT nodes. Table 6 presents the statistics of these variants, including number of entities, number of relations, and number of triples. The column "CVT" indicates whether each dataset includes or excludes CVT nodes, and the column "reverse" indicates whether the dataset includes or excludes reverse triples. Correspondingly, the dataset names use +/− of CVT/REV to denote these characteristics. The type system we created is also provided as auxiliary information. Metadata and administrative triples are removed, and thus the variants only include subject matter triples. The rest of this section provides details about how the variants were created from the original Freebase data dump, which is nontrivial largely due to the scarcity of available documentation.

Table 6. Statistics of the four variants of Freebase

Variant	CVT	reverse	#Entities	#Relations	#Triples
FB-CVT-REV	×	×	46,069,321	3,055	125,124,274
FB-CVT+REV	×	✓	46,077,533	5,028	238,981,274
FB+CVT-REV	✓	×	59,894,890	2,641	134,213,735
FB+CVT+REV	✓	✓	59,896,902	4,425	244,112,599

URI Simplification. In a Freebase triple (`subject`, *predicate*, `object`), each component that is not a literal value is identified by a URI (uniform resource identifier) [25]. For simplification and usability, we removed URI prefixes such as *"<http://rdf.freebase.com/>"*, *"<http://rdf.freebase.com/ns/>"* and *"<http://www.w3.org/[0-9]*/[0-9]*/[0-9]*-*>"*. We only retained URI segments corresponding to domains, types, properties' labels, and MIDs. These segments are dot-delimited in the URI. For better readability, we replaced the dots by *"/"*. For example, URI *<http://rdf.freebase.com/ns/film.director.film>* is simplified to */film/director/film*. Likewise, *<http://rdf.freebase.com/ns/award.award_winner>* and *<http://rdf.freebase.com/ns/m.0zbqpbf>*, which are the URIs of a Freebase type and an MID, are simplified to */award/award_winner* and `/m/0zbqpbf`. The mapping between original URIs and simplified labels are also included in our datasets as auxiliary information.

Extracting Metadata. The non-subject matter triples are used to extract metadata about the subject matter triples. We created a mapping between Freebase entities and their types using predicate */type/object/types*. Using predicate */type/object/name*, we created a lookup table mapping the MIDs of entities to their labels. Similarly, using predicate */type/object/id*, we created lookup tables mapping MIDs of Freebase domains, types and properties to their labels.

Detecting Reverse Triples. As discussed in Sect. 3, Freebase has a property */type/property/reverse_property* for denoting reverse relations. A triple (`r1`, */type/property/reverse_property*, `r2`) indicates that relations *r1* and *r2* are reverse of each other. When we remove reverse triples to produce FB-CVT-REV and FB+CVT-REV, i.e., triples belonging to reverse relations, we discard all triples in relation *r2*.

Detecting Mediator Nodes. Our goal is to identify and separate all mediator (CVT) nodes. It is nontrivial as Freebase does not directly denote CVT nodes although it does specify 2,868 types as *mediator types*. According to our empirical analysis, a mediator node can be defined as a Freebase object that belongs to at least one mediator type but was given no label. One example is object `/m/011tzbfr` which belongs to the mediator type */comedy/comedy_group_membership* but has no label. Once we found all CVTs, we created Freebase variants with and without such nodes. The variants without CVTs were produced by creating concatenated edges that collapse CVTs and merge intermediate edges (edges with at least one CVT endpoint). For instance, the triples (`BAFTA Award for Best Film`, */award/award_category/nominees*, `CVT`) and (`CVT`, */award/award_nomination/award_nominee*, `James Ivory`) in Fig. 1 would be concatenated to form a new triple (`BAFTA Award for Best Film`, */award/award_category/nominees–/award/award_nomination/award_nominee*, `James Ivory`). Note that, in converting n-ary relationships to binary relationships, the concatenation does not need to be carried out along edges in the same direction. For each pair of reverse triples only one is kept, and the choice of which

one to keep is random. Two edges connected to the same CVT node thus can have various combinations of directions, depending how their reverse edges were randomly removed. Moreover, the performance of the models cannot be affected by these random selection of reverse triple removal.

7 Experiments

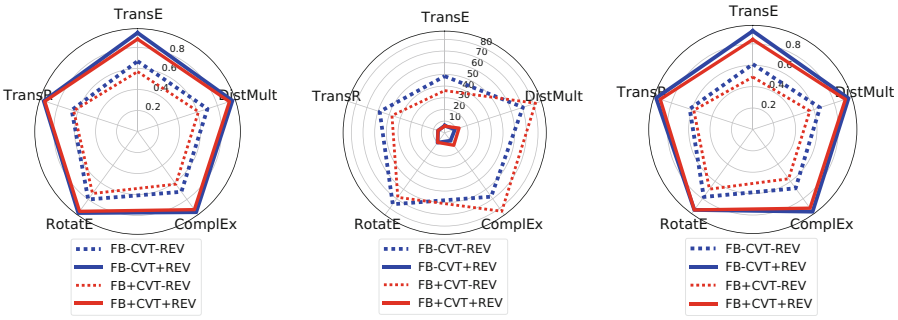
Task. The *link prediction* task as described in [10] is particularly widely used for evaluating different embedding methods. Its goal is to predict the missing `h` or `t` in a triple (`h`, r, `t`). For each test triple (`h`, r, `t`), the head entity `h` is replaced with every other entity `h'` in the dataset, to form *corrupted* triples. The original test triple and its corresponding corrupted triples are ranked by their scores according to a scoring function. The scoring function takes learned entity and relation representations as input. The rank of the original test triple is denoted $rank_{\texttt{h}}$. The same procedure is used to calculate $rank_{\texttt{t}}$ for the tail entity `t`. A method with the ideal performance should rank the test triple at top.

Evaluation Measures. We gauge the accuracy of embedding models by several commonly used measures in [10] and follow-up studies, including `Hits@1`$^{\uparrow}$, `Hits@3`$^{\uparrow}$, `Hits@10`$^{\uparrow}$, `MR`$^{\downarrow}$ (Mean Rank), and `MRR`$^{\uparrow}$ (Mean Reciprocal Rank). An upward/downward arrow beside a measure indicates that methods with greater/smaller values by that measure possess higher accuracy. Instead of directly using the above-mentioned raw metrics', we use their corresponding *filtered* metrics [10], denoted `FHits@1`$^{\uparrow}$, `FHits@3`$^{\uparrow}$, `FHits@10`$^{\uparrow}$, `FMR`$^{\downarrow}$, and `FMRR`$^{\uparrow}$. In calculating these measures, corrupted triples that are already in training, test or validation sets do not participate in ranking. In this way, a model is not penalized for ranking other correct triples higher than a test triple.

Models. We trained and evaluated five well-known link prediction embedding models—TransE [10], TransR [27], DistMult [48], ComplEx [40], and RotatE [38]—on the four variant datasets of Freebase discussed in Sect. 6. TransE, RotatE and TransR are three representative translational distance models. DistMult and ComplEx are semantic matching models that exploit similarity-based scoring functions [51].

Experiment Setup. Multi-processing, multi-GPU distributed training frameworks have recently become available to scale up embedding models [26,51,52]. Our experiments were conducted using one such framework, DGL-KE [51], with the settings and hyperparameters suggested in [51]. The experiments used an Intel-based machine with a Xeon E5-2695 processor running at 2.1 GHz, Nvidia Geforce GTX1080Ti GPU, and 256 GB RAM. The datasets were randomly divided into training, validation and test sets with the split ratio of 90/5/5, as in [51]. In our two datasets with CVT nodes, we made sure that a CVT node present in the test or validation set is also present in the training set. More details

on experiment setup as well as training and inference time logs are available from our GitHub repository.



(a) MRR$^{\uparrow}$ (b) MR$^{\downarrow}$ (c) Hits@1$^{\uparrow}$

Fig. 5. Link prediction performance on our four new variants of Freebase

Results on Full-Scale vs. Small-Scale Freebase Datasets. The experiment results are reported in Table 7 and Fig. 5. Link prediction results on full-scale Freebase datasets have never been reported before, barring results on problematic datasets such as Freebase86m which we explained in Sect. 3.3. Our datasets FB-CVT-REV and FB-CVT+REV can be viewed as the full-scale counterparts of FB15k-237k and FB15k (of which the results are in Table 3), respectively. Comparing the results on the full-scale and small-scale datasets shows that models have much stronger performance on the full-scale datasets. Our goal is not to compare different models or optimize the performance of any particular model. Rather, the significant performance gap between the full-scale and small-scale Freebase datasets is worth observing and not reported before. This accuracy difference could be attributed to the dataset size difference, as is the case in machine learning in general. Results like these suggest that our datasets can provide opportunities to evaluate embedding models more realistically.

Impact of Reverse Relations. The impact of reverse relations at the scale of the full Freebase dataset was never studied before. This paper thus fills the gap. As Fig. 5 and Table 7 show, results on the two variants without CVT nodes—FB-CVT-REV (reverse relations excluded) and FB-CVT+REV (reverse relations included)—present substantial over-estimation of link prediction models' accuracy when reverse triples are included. So do the results on the two variants with CVT nodes—FB+CVT-REV and FB+CVT+REV.

Impact of Mediator Nodes. As articulated in Sect. 3.2, no prior work has studied the impact of mediator nodes on link prediction, regardless of dataset

Table 7. Link prediction performance on four new Freebase variants and Freebase86m

	FB-CVT-REV				
Model	MRR↑	MR↓	Hits@1↑	Hits@3↑	Hits@10↑
TransE	0.67	48.49	0.61	0.70	0.78
DistMult	0.70	70.49	0.66	0.72	0.77
ComplEx	0.71	67.74	0.68	0.73	0.78
TransR	0.66	58.55	0.62	0.68	0.74
RotatE	0.80	75.72	0.78	0.81	0.84
	FB-CVT+REV				
Model	MRR↑	MR↓	Hits@1↑	Hits@3↑	Hits@10↑
TransE	0.94	6.07	0.92	0.95	0.97
DistMult	0.95	9.23	0.94	0.96	0.97
ComplEx	0.95	8.43	0.95	0.96	0.97
TransR	0.94	5.98	0.93	0.95	0.96
RotatE	0.96	10.43	0.95	0.96	0.97
	FB+CVT-REV				
Model	MRR↑	MR↓	Hits@1↑	Hits@3↑	Hits@10↑
TransE	0.57	36.12	0.49	0.61	0.75
DistMult	0.61	81.84	0.56	0.63	0.70
ComplEx	0.62	83.20	0.57	0.64	0.70
TransR	0.64	47.52	0.58	0.66	0.75
RotatE	0.73	68.43	0.69	0.75	0.80
	FB+CVT+REV				
Model	MRR↑	MR↓	Hits@1↑	Hits@3↑	Hits@10↑
TransE	0.88	5.60	0.84	0.92	0.96
DistMult	0.92	12.92	0.91	0.93	0.95
ComplEx	0.92	13.27	0.91	0.93	0.95
TransR	0.93	6.07	0.91	0.94	0.96
RotatE	0.94	10.26	0.93	0.95	0.96
	Freebase86m				
Model	MRR↑	MR↓	Hits@1↑	Hits@3↑	Hits@10↑
TransE	0.72	23.27	0.65	0.77	0.87
DistMult	0.83	45.54	0.81	0.84	0.87
ComplEx	0.83	46.55	0.81	0.84	0.86
TransR	0.65	71.91	0.61	0.68	0.74
RotatE	0.82	65.46	0.81	0.82	0.84

Table 8. Triple classification results on FB15k-237

	consistent h			
Model	Precision	Recall	Acc	F1
TransE	0.52	0.59	0.52	0.55
DistMult	0.53	0.51	0.53	0.52
ComplEx	0.54	0.48	0.53	0.51
RotatE	0.52	0.53	0.52	0.52
	inconsistent h			
Model	Precision	Recall	Acc	F1
TransE	0.81	0.69	0.76	0.74
DistMult	0.94	0.87	0.91	0.90
ComplEx	0.94	0.88	0.91	0.91
RotatE	0.89	0.83	0.87	0.86
	consistent t			
Model	Precision	Recall	Acc	F1
TransE	0.58	0.54	0.57	0.56
DistMult	0.59	0.55	0.58	0.57
ComplEx	0.60	0.56	0.59	0.58
RotatE	0.60	0.47	0.58	0.53
	inconsistent t			
Model	Precision	Recall	Acc	F1
TransE	0.90	0.82	0.86	0.86
DistMult	0.95	0.89	0.92	0.92
ComplEx	0.95	0.90	0.93	0.92
RotatE	0.87	0.78	0.83	0.82

scale. Comparing the results on the two variants without reverse triples—FB-CVT-REV (mediator nodes excluded) and FB+CVT-REV (mediator nodes included), as illustrated in Fig. 5 and Table 7, shows that the existence of CVT nodes led to weaker model accuracy. Although the results on FB-CVT+REV and FB+CVT+REV are over-estimations since they both retained reverse triples, similar observation regarding mediator nodes is still made—the models are slightly less accurate on FB+CVT+REV (mediator nodes included) than FB-CVT+REV (mediator nodes excluded). More detailed analyses remain to be done, in order to break down different impacts of individual factors that contribute to the performance degeneration, such as the factors analyzed in Sect. 3.2. Our newly created datasets will facilitate research in this direction.

Freebse86m vs. FB+CVT+REV. Comparing the results on these two datasets, as shown in Table 7 and Fig. 4, reveals that the existence of non-subject matter triples degenerates model performance. In general, non-subject matter triples and subject-matter triples should be examined separately given their fundamental difference. Mixing them together hinders robust understanding of embedding models' effectiveness in predicting knowledge facts.

Usefulness of the Type System. To demonstrate the usefulness of the Freebase type system we created (Sect. 4), we evaluated embedding models' performance on the task of triple classification [43] using the LibKGE library [11]. This task is the binary classification of triples regarding whether they are true or false facts. We needed to generate a set of negative triples in order to conduct this task. The type system proves useful in generating type-consistent negative samples. When triple classification was initially used for evaluating models [36,43], negative triples were generated by randomly corrupting head or tail entities of test and validation triples. The randomly generated negative test cases are not challenging as they mostly violate type constraints which were discussed in Sect. 4, leading to overestimated classification accuracy. Pezeshkpour et al. [32] and Safavi et al. [35] noted this problem and created harder negative samples. Inspired by their work, we created two sets of negative samples for test and validation sets of FB15k-237. One set complies with type constraints and the other violates such constraints. To generate a type consistent negative triple for a test triple (`h`, *r*, `t`), we scan the ranked list generated for tail entity prediction to find the first entity `t'` in the list that has the expected type for the objects of relation *r*. We then add the corrupted triple (`h`, *r*, `t'`) to the set of type consistent negative triples for tail entities if it does not exists in FB15k-237. We repeat the same procedure to corrupt head entities and to create negative samples for validation data. To generate type-violating negative triples we just make sure the type of the entity used to corrupt a positive triple is different from the original entity's type. The results of triple classification on these new test sets are presented in Table 8. Note that Table 8 does not include TransR since it is not implemented in LibKGE. The results in the table show that the models' performance on type-consistent negative samples are much lower than their performance on type-violating negative samples.

8 Conclusion

We laid out a comprehensive analysis of the challenges associated with Freebase data modeling idiosyncrasies, including CVT nodes, reverse properties, and type system. To tackle these challenges, we provide four variants of the Freebase dataset by inclusion and exclusion of these idiosyncrasies. We further conducted experiments to evaluate various link prediction models on these datasets. The results fill an important gap in our understanding of embedding models for knowledge graph link prediction as such models were never evaluated using a proper full-scale Freebase dataset. The paper also fills an important gap in

dataset availability as this is the first-ever publicly available full-scale Freebase dataset that has gone through proper preparation.

Acknowledgments. This material is based upon work supported by the National Science Foundation under Grants IIS-1719054 and IIS-1937143.

Resource Availability Statement. The code and experiment results produced from this work are available from GitHub [1] and the datasets are made available at Zenodo [2].

References

1. The GitHub repository of Freebases (2022). https://github.com/idirlab/freebases
2. Freebase datasets for robust evaluation of knowledge graph link prediction models (2023). https://doi.org/10.5281/zenodo.7909511
3. Open knowledge network roadmap: powering the next data revolution (2023). https://new.nsf.gov/tip/updates/nsf-releases-open-knowledge-network-roadmap-report
4. Akrami, F., Guo, L., Hu, W., Li, C.: Re-evaluating embedding-based knowledge graph completion methods. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 1779–1782. Association for Computing Machinery, Turin, Italy (2018)
5. Akrami, F., Saeef, M.S., Zhang, Q., Hu, W., Li, C.: Realistic re-evaluation of knowledge graph completion methods: an experimental study. In: Proceedings of the 2020 ACM Special Interest Group on Management of Data International Conference on Management of Data, pp. 1995–2010. Association for Computing Machinery, Portland, Oregon, USA (2020)
6. Allemang, D., Hendler, J.: Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL. Elsevier, Online (2011)
7. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: a nucleus for a web of open data. In: Aberer, K., et al. (eds.) Proceedings of the 6th International Semantic Web Conference and 2nd Asian Semantic Web Conference. LNCS, vol. 4825, pp. 722–735. Springer, Cham (2007). https://doi.org/10.1007/978-3-540-76298-0_52
8. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM Special Interest Group on Management of Data International Conference on Management of Data, pp. 1247–1250. Association for Computing Machinery, Vancouver, Canada (2008)
9. Bollacker, K., Tufts, P., Pierce, T., Cook, R.: A platform for scalable, collaborative, structured information integration. In: International Workshop on Information Integration on the Web, pp. 22–27. AAAI Press, Vancouver, British Columbia (2007)
10. Bordes, A., Usunier, N., Garcia-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Proceedings of the 26th International Conference on Neural Information Processing Systems, pp. 2787–2795. Curran Associates, Lake Tahoe, Nevada, United States (2013)

11. 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. Association for Computational Linguistics, Online (2020)
12. Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka, E.R., Mitchell, T.M.: Toward an architecture for never-ending language learning. In: Twenty-Fourth AAAI Conference on Artificial Intelligence, pp. 1306–1313. AAAI Press, New York, USA (2010)
13. Chah, N.: Freebase-triples: a methodology for processing the Freebase data dumps. arXiv preprint arXiv:1712.08707 (2017)
14. Code, G.: Wikipedia links data (2012). https://code.google.com/archive/p/wikilinks/. Accessed 9 June 2022
15. Dettmers, T., Pasquale, M., Pontus, S., Riedel, S.: Convolutional 2D knowledge graph embeddings. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence, pp. 1811–1818. AAAI Press, New Orleans, Louisiana, USA (2018)
16. Färber, M.: Semantic Search for Novel Information. IOS Press, Amsterdam (2017)
17. Ferrucci, D., et al.: Building Watson: an overview of the DeepQA project. AI Mag. **31**(3), 59–79 (2010)
18. Google: Freebase data dumps (2013). https://developers.google.com/freebase. Accessed 11 Nov 2022
19. Grant, J., Beckett, D.: RDF test cases (2004). https://www.w3.org/TR/rdf-testcases/
20. Guan, S., Jin, X., Wang, Y., Cheng, X.: Link prediction on N-ary relational data. In: The World Wide Web Conference, pp. 583–593. Association for Computing Machinery, San Francisco, CA, USA (2019)
21. Guo, S., Wang, Q., Wang, B., Wang, L., Guo, L.: Semantically smooth knowledge graph embedding. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pp. 84–94. Association for Computational Linguistics, Beijing, China (2015)
22. Hao, Y., et al.: An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pp. 221–231. Association for Computational Linguistics, Vancouver, Canada (2017)
23. Hu, W., et al.: Open graph benchmark: datasets for machine learning on graphs. Adv. Neural Inf. Process. Syst. **33**, 22118–22133 (2020)
24. Ji, S., Pan, S., Cambria, E., Marttinen, P., Philip, S.Y.: A survey on knowledge graphs: representation, acquisition, and applications. IEEE Trans. Neural Networks Learn. Syst. **33**(2), 494–514 (2021)
25. Klyne, G.: Resource description framework (RDF): Concepts and abstract syntax (2004). http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/
26. Lerer, A., et al.: PyTorch-BigGraph: a large scale graph embedding system. Proc. Mach. Learn. Syst. **1**, 120–131 (2019)
27. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Proceedings of the 29th AAAI Conference on Artificial Intelligence, pp. 2181–2187. AAAI Press, Austin, Texas, USA (2015)
28. Mohoney, J., Waleffe, R., Xu, H., Rekatsinas, T., Venkataraman, S.: Marius: learning massive graph embeddings on a single machine. In: 15th USENIX Symposium on Operating Systems Design and Implementation, pp. 533–549. The Advanced Computing Systems Association, Online (2021)

29. Networking, information technology research, development: Open knowledge network: Summary of the big data IWG workshop (2018). https://www.nitrd.gov/open-knowledge-network-summary-of-the-big-data-iwg-workshop/
30. Noy, N., Gao, Y., Jain, A., Narayanan, A., Patterson, A., Taylor, J.: Industry-scale knowledge graphs: lessons and challenges. Commun. ACM **62**(8), 36–43 (2019)
31. Pellissier Tanon, T., Vrandečić, D., Schaffert, S., Steiner, T., Pintscher, L.: From Freebase to Wikidata: the great migration. In: Proceedings of the 25th International Conference on World Wide Web, pp. 1419–1428. Association for Computing Machinery, Montreal, Canada (2016)
32. Pezeshkpour, P., Tian, Y., Singh, S.: Revisiting evaluation of knowledge base completion models. In: Automated Knowledge Base Construction, p. 10. OpenReview, Online (2020)
33. Rossi, A., Barbosa, D., Firmani, D., Matinata, A., Merialdo, P.: Knowledge graph embedding for link prediction: a comparative analysis. ACM Trans. Knowl. Discov. Data (TKDD) **15**(2), 1–49 (2021)
34. Rosso, P., Yang, D., Cudré-Mauroux, P.: Beyond triplets: hyper-relational knowledge graph embedding for link prediction. In: Proceedings of The Web Conference 2020, pp. 1885–1896. Association for Computing Machinery, Online (2020)
35. Safavi, T., Koutra, D.: CoDEx: a comprehensive knowledge graph completion benchmark. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pp. 8328–8350. Association for Computational Linguistics, Online (2020)
36. Socher, R., Chen, D., Manning, C.D., Ng, A.: Reasoning with neural tensor networks for knowledge base completion. In: Proceedings of the 26th International Conference on Neural Information Processing Systems, pp. 926–934. Curran Associates, Lake Tahoe, United States (2013)
37. Suchanek, F.M., Kasneci, G., Weikum, G.: YAGO: a large ontology from Wikipedia and WordNet. J. Web Semant. **6**(3), 203–217 (2008)
38. Sun, Z., Deng, Z.H., Nie, J.Y., Tang, J.: RotatE: knowledge graph embedding by relational rotation in complex space. In: Proceedings of the International Conference on Learning Representations, pp. 926–934. OpenReview.net, New Orleans, LA, USA (2019)
39. 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, Beijing, China (2015)
40. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: Proceedings of the 33rd International Conference on Machine Learning, pp. 2071–2080. JMLR.org, New York City, NY, USA (2016)
41. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledge base. Commun. ACM **57**(10), 78–85 (2014)
42. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: a survey of approaches and applications. IEEE Trans. Knowl. Data Eng. **29**(12), 2724–2743 (2017)
43. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the 28th AAAI Conference on Artificial Intelligence, pp. 1112–1119. AAAI Press, Québec City, Québec, Canada (2014)
44. Wen, J., Li, J., Mao, Y., Chen, S., Zhang, R.: On the representation and embedding of knowledge bases beyond binary relations. In: Proceedings of the International

Joint Conference on Artificial Intelligence, pp. 1300–1307. IJCAI, New York City, USA (2016)
45. Xie, R., Liu, Z., Sun, M., et al.: Representation learning of knowledge graphs with hierarchical types. In: Proceedings of International Joint Conference on Artificial Intelligence, vol. 2016, pp. 2965–2971. IJCAI, New York City, USA (2016)
46. Xiong, C., Power, R., Callan, J.: Explicit semantic ranking for academic search via knowledge graph embedding. In: Proceedings of the 26th International Conference on World Wide Web, pp. 1271–1279. Association for Computing Machinery, Perth, Australia (2017)
47. Yang, B., Mitchell, T.: Leveraging knowledge bases in LSTMs for improving machine reading. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (vol. 1: Long Papers), pp. 1436–1446. Association for Computational Linguistics, Vancouver, Canada (2017)
48. Yang, B., Yih, 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, p. 12. OpenReview.net, San Diego, CA, USA (2015)
49. Zhang, F., Yuan, N.J., Lian, D., Xie, X., Ma, W.Y.: Collaborative knowledge base embedding for recommender systems. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 353–362. Association for Computing Machinery, San Francisco, CA, USA (2016)
50. Zhang, R., Li, J., Mei, J., Mao, Y.: Scalable instance reconstruction in knowledge bases via relatedness affiliated embedding. In: Proceedings of the 2018 World Wide Web Conference, pp. 1185–1194. Association for Computing Machinery, Lyon, France (2018)
51. Zheng, D., et al.: DGL-KE: training knowledge graph embeddings at scale. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 739–748. Association for Computing Machinery, Xi'an, China (2020)
52. Zhu, Z., Xu, S., Qu, M., Tang, J.: GraphVite: a high-performance CPU-GPU hybrid system for node embedding. In: The World Wide Web Conference, pp. 2494–2504. Association for Computing Machinery, San Francisco, CA, USA (2019)