Comprehensive Analysis of Freebase and Creation of Datasets for Robust Evaluation of Knowledge Graph Completion Methods [Experiment, Analysis & Benchmark]

Nasim Shirvani-Mahdavi*
University of Texas at Arlington
nasim.shirvanimahdavi2@mavs.uta.edu

Farahnaz Akrami*
University of Texas at Arlington
farahnaz.akrami@mavs.uta.edu

Mohammed Samiul Saeef*
University of Texas at Arlington
mohammedsamiul.saeef@mavs.uta.edu

Xiao Shi
University of Texas at Arlington
xiao.shi@mavs.uta.edu

Chengkai Li
University of Texas at Arlington
cli@uta.edu

ABSTRACT

Knowledge graphs are an essential asset to a wide variety of applications. 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. Specifically, when models are developed and evaluated agnostically of these idiosyncrasies, one could either miss the opportunity to leverage such features or fall into pitfalls without knowing. One example is that many knowledge graph link prediction models proposed in the past decade were evaluated using a subset of Freebase full of reverse triple pairs. The reverse triples lead to data leakage in evaluating the models. The consequence is substantial overestimation of the models' accuracy and faulty comparison of their relative strengths. 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.

PVLDB Reference Format:
Nasim Shirvani-Mahdavi*, Farahnaz Akrami*, Mohammed Samiul Saeef*, Xiao Shi, and Chengkai Li. Comprehensive Analysis of Freebase and Creation of Datasets for Robust Evaluation of Knowledge Graph Completion Methods [Experiment, Analysis & Benchmark]. PVLDB, 14(1): XXX-XXX, 2023.
doi:XX.XX/XXX.XX

* Equal contribution

Proceedings of the VLDB Endowment, Vol. 14, No. 1 ISSN 2150-8097.
doi:XX.XX/XXX.XX

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/idirlab/freebases.

1 INTRODUCTION

Knowledge graphs (KGs) encode semantic, factual information as triples of the form (subject s, predicate *p*, object o). They can potentially link together heterogeneous data sources across different domains for purposes greater than what they support separately. This makes KGs an essential asset to a wide variety of tasks and applications in the fields of artificial intelligence and machine learning [10, 24], including natural language processing [53], search [52], question answering [21], and recommender systems [55]. Consequently, KGs are of great importance to many technology companies [16, 31] and governments [30].

To develop and robustly evaluate models and algorithms for tasks on knowledge graphs, access to large-scale KGs is crucial. But publicly available KG datasets are often much smaller than what real-world scenarios render and require [22]. For example, FB15k and FB15k-237 [7, 42], 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 knowledge graphs are both large and publicly available, including DBpedia [4], Freebase [5], Wikidata [44], YAGO [40], and NELL [9].

With more than 80 million nodes, Freebase is amongst the largest public KGs. It comprises factual information in a broad range of domains, making it relevant to many applications. 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 [32]. *Secondly*, Freebase uses mediator objects to facilitate representation of *n*-ary relationships [32]. *Lastly*, Freebase's strong de facto type system categorizes each entity into one or more types, and the type of an entity determines the properties it may possess [6]. Furthermore, in practice the label of a property *almost* functionally determines the types of the entities in 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 [34, 46] 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 [2].

This paper lays out a comprehensive analysis of the challenges associated with the aforementioned idiosyncrasies of Freebase. It measures their impact on knowledge graph embedding models, and it provides four variants of the Freebase dataset by inclusion/exclusion of mediator objects and reverse triples. A Freebase type system is also extracted to supplement the variants. Furthermore, the datasets underwent thorough cleaning in order to improve their utility and to remove irrelevant triples from the original Freebase data dump [17]. 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 [57], 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. (Details in Section 5.)

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 (e.g., the aforementioned Freebase86m) used were problematic, leading to unreliable results. The experiments on our datasets, reported in Section 7, 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, instead of unreliable results; 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. We examined all full-length publications that 1) appeared in 12 top conferences during their latest years and 2) used datasets commonly used for link prediction. This amounts to 53 publications. [1] Among all these publications, 48 publications used datasets produced from Freebase, while 8 used datasets from Wikidata. Only 3 publications used a Freebase dataset at its full scale, specifically Freebase86m. This suggests that researchers may not be able to carry out large-scale study due to lack of proper datasets. For this reason, although this paper focuses on Freebase's usage in knowledge graph completion, our new datasets are valuable to researcher and practitioners in various other tasks and applications.

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 Sections 3 and 4, was defined or detailed in the data dump itself. Figuring out the details in these sections required iterative trial-and-error in examining the data. In fact, we are unaware of more detailed description of these idiosyncrasies 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. Given the datasets and experiment results made available in this paper, it becomes possible to compare the real performance of conventional embedding models (e.g., TransE [7] and ComplEx [43]) and hyper-relational fact models [19, 36, 48, 56] on a full-scale Freebase dataset that includes multiary relationships (i.e., mediator objects). Our results on the subject matter triples demonstrate the performance of triples containing mediator objects (i.e., multiary relationships) and the binary relationships separately and overall. These results can be compared with the results of hyper-relational fact models on the same dataset in which the multiary relationships are converted to hyper facts. Such a comparison will be the first not only for Freebase but also for any large-scale knowledge graph. The experiment design could be similarly extended to study the impact of multiary relationships in Wikidata on embedding models and compare such models with models built for hyper-relational facts. Details related to this can be found in Section 4.2.

2 FREEBASE BASIC CONCEPTS

This section provides a brief summary of some basic terminology and concepts related to Freebase. We aim to be consistent with [6, 18, 25, 32] in nomenclature and notation.

RDF: Freebase is available from its data dumps [17] 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`, `A Room with a View`, and `BAFTA Award for Best Film` in Figure 1. An example of non-topic nodes is CVT (Compound Value Type) nodes which are used to represent *n*-ary relations (details in Section 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/041d94`. 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.

[1] The conferences, the papers, and the datasets used in the papers are listed in file "papers.xlsx" which can be accessed at the top directory of our GitHub repository https://github.com/idirlab/freebases.

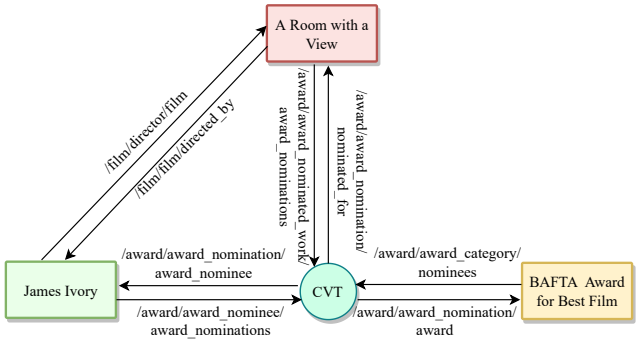


Figure 1: A small fragment of Freebase, with a mediator node

Type: 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 Figure 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 uniquely 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.

3 USEFUL IDIOSYNCRASIES OF FREEBASE

Freebase is amongst the largest cross-domain common fact knowledge graphs that is publicly available. 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 [10]. As stated in [15], Freebase's data is consistent, semantically valid, and certified free of error to a very admissible degree. Before Google shut down Freebase in 2015, the company announced its plan to help with the transfer of Freebase content to Wikidata [13]. This transfer is yet to be completed [32, 50]. Nevertheless, Freebase remains the single most commonly used dataset for the task of link prediction, as mentioned in Section 1. This section explains several idiosyncrasies of Freebase's data modeling design choices.

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* [14, 32]. 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`). Hence, (`A Room With A View`, */film/film/directed_by*, `James Ivory`) and (`James Ivory`, */film/director/film*, `A Room With A View`) form reverse triples, shown as two edges in reverse directions in Figure 1.

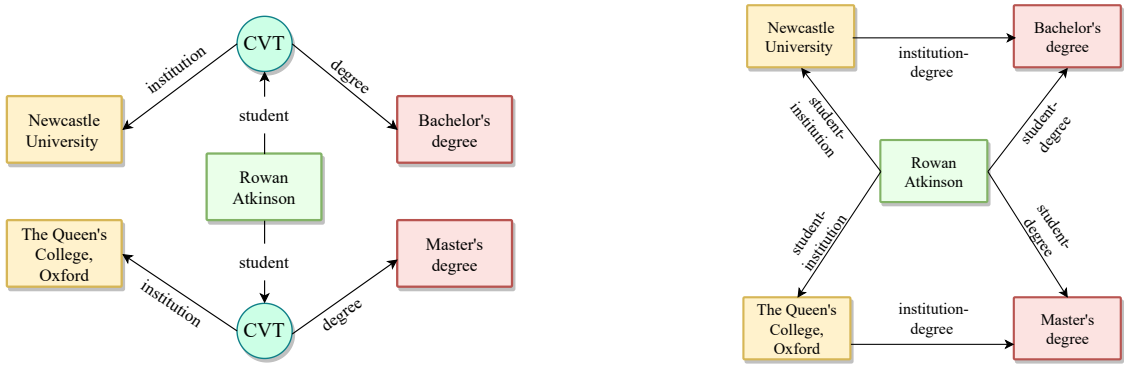
Mediator Nodes *Mediator nodes*, also called CVT nodes, are used in Freebase to represent n-ary relationships [32]. For example, Figure 1 shows a CVT node connected to an `award`, a `nominee`, and a `work`. This or similar approach is necessary for accurate modeling of the real-world. However, this modeling complexity presents a non-trivial challenge to tasks such as link prediction. As we shall discuss in Section 4 and Section 7, the presence of mediator nodes decreases the accuracy of link prediction embedding models.

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. But this only helps reduce the complexity of algorithmic solutions. Such a transformation in fact leads to loss of information [48] and is irreversible [35], and thus it may not always be an acceptable approach as far as data semantics is concerned. For better clarity, we use Figure 2 to further illustrate the loss of information. [2] Figure 2a depicts two ternary relationships about `Rowan Atkinson`'s educational institutions and his degrees. Figure 2b is after transforming the ternary relationships into binary relationships. As the figures show, the transformed graph could not exactly indicate that `Rowan Atkinson` studied which degree in which institution.

Nevertheless, most prior studies of knowledge graph link prediction use Freebase datasets without CVT nodes, e.g., FB15k and FB15k-237 in which the concatenation mentioned above was carried out. Though lossful for Freebase-like knowledge graphs, as explained above, the insights gained could be more applicable toward graphs with only binary relationships.

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. In Figure 2a, 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. The performance of the models cannot be affected by these random selection of reverse triple removal.

[2] For simplicity of presentation, we use [label] instead of /[domain]/[type]/[label] to denote edge labels. In the post-transformation graph, the label of a concatenated edge is the concatenation of the two original edge labels.



(a) Before transformation: n-ary relationships modeled via CVT nodes **(b) After transformation: binary relationships without CVT nodes**

Figure 2: An example of information loss after converting n-ary relationships to binary relationships

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 Section 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" [6]. 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 Section 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, $P(o \in \textit{film/actor}) = 0.9969$, $P(o \in \textit{tv/tv\_actor}) = 0.1052$, $P(o \in \textit{music/artist}) = 0.0477$, $P(o \in \textit{award/award\_winner}) = 0.0373$, and $P(o \in \textit{people/person}) = 0.998$. 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, $P(o \in \textit{people/person} \mid o \in \textit{film/actor}) = 0.9984$ and $P(o \in \textit{film/actor} \mid o \in \textit{people/person}) = 0.1394$. Thus, we assigned *film/actor* as the required entity type for the object node of edge type *film/film/performance* because it is more specific than *people/person*, even though *people/person* had slightly higher coverage.

The threshold $\alpha = 0.95$ was chosen based on empirical evidence, as it yielded better accuracy than other α values we tried. Given the instances of an edge type, the accuracy of an α value is measured by the percentage of a subject or object node n actually belonging to the type t assigned based on α, i.e., the $P(n \in t)$. The average type assignment accuracy scores for node instances (both subject and object nodes) of all edge types are 0.93707, 0.99134, 0.99413, and 0.99692, for $\alpha = 0.5$, $\alpha = 0.85$, $\alpha = 0.9$, and $\alpha = 0.95$, respectively. This verifies the chosen threshold $\alpha = 0.95$ as a strong choice. We also observed the impact of different values of α on the type system for a specific edge type. For instance, given edge type */organization/organization/place_founded*, the types assigned to the object node were */olympics/olympic_participating_country*, */location/dated_location* and */location/location* for $\alpha = 0.5$, $\alpha = 0.85$ and $\alpha = 0.95$, respectively. The assigned type provides better coverage (but still being sufficiently specific) as we increase α. The most appropriate type in this example is obtained at $\alpha = 0.95$, based on manual inspection.

The type system we created can be useful in improving link prediction. A few studies in fact employed type information for such a goal [20, 51]. Particularly, embedding models can aim to keep entities of the same type close to each other in the embedding space [20]. 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*.

4 CHALLENGES POSED BY FREEBASE IDIOSYNCRASIES

As mentioned earlier, the data modeling idiosyncrasies of Freebase could pose challenges that hamper the advancement of knowledge graph oriented technologies. This section explains such challenges and their specific impacts on link prediction task.

4.1 Reverse Triples

Several previous studies discussed the pitfalls in including reverse relations (as discussed in Section 3) in datasets used for knowledge graph link prediction task [1, 2, 12, 42]. Link prediction is the task of predicting the missing s in triple (?, p, o) or missing o in (s, p, ?). The popular benchmark dataset FB15k (a relatively small subset of Freebase), created by Bordes et al. [7], was almost always used for this task. Toutanova and Chen [42] 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 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 [2], 2) Instead of complex models, one may achieve similar results by using statistics of the triples to derive simple rules of the form (s, p_1, o) $\Rightarrow$ (o, p_2, s) which are highly effective given the prevalence of reverse relations [2, 12], and 3) The link prediction scenario, given such data, is non-existent in the real-world at all. With regard to FB15k, the redundant reverse relations, coming from Freebase, were just artificially created. As mentioned in Section 3, new facts were added into Freebase as pairs of reverse triples, and reverse relations were denoted explicitly by the relation */type/property/reverse_property* [14, 32]. 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. Training a knowledge graph completion model using FB15k is thus a form of *overfitting* in that the learned model is optimized for the reverse triples which cannot be generalized to realistic settings. 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 prediction.

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.

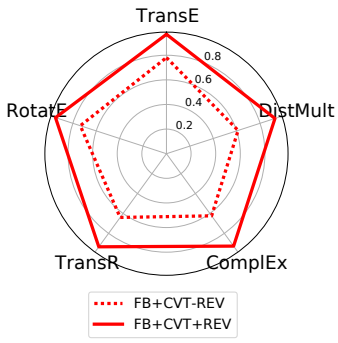


Figure 3: Impact of reverse triples on the performance (MRR$^{\uparrow}$) of embedding models

The impact of reverse triples was previously only examined on small-scale datasets FB15k and FB15k-237 [1, 2, 12, 42]. 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 Section 6 (specifically Table 7 and Figure 6), here we summarize the most important observations. Figure 3 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 Section 6). FB+CVT+REV (the counterpart of the small-scale FB15k) is obtained after cleaning the Freebase data dump and removing irrelevant data, and in FB+CVT-REV (the counterpart of the small-scale FB15k-237) reverse relations are further removed by only keeping one relation out of each reverse pair. In the radar chart of Figure 3, the solid and dashed red polygons depict various models' MRR$^{\uparrow}$ on FB+CVT+REV and FB+CVT-REV, respectively. Similar to the comparison 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.

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

To put the results in context, we reproduced results on FB15k and FB15k-237 using DGL-KE [57], which is the framework we used in this study for experiments on large-scale datasets. The results are in Table 2. They are mostly consistent with previously reported results using frameworks (e.g., LibKGE [8]) for small-scale datasets, barring differences that can be attributed to implementations of different frameworks. Comparing Table 1 and Table 2, we can observe that models' performance on full-scale datasets is significantly higher

Table 1: Link prediction performance (MRR↑) on FB+CVT-REV and FB+CVT+REV

	FB+CVT-REV			FB+CVT+REV		
Model	**unidirectional**	**bidirectional**	**all**	**unidirectional**	**bidirectional**	**all**
TransE	0.883	0.771	0.781	0.910	0.974	0.970
DistMult	0.656	0.607	0.612	0.707	0.940	0.927
ComplEx	0.674	0.619	0.624	0.698	0.942	0.928
TransR	0.668	0.637	0.640	0.754	0.946	0.935
RotatE	0.679	0.741	0.736	0.733	0.961	0.948

Table 2: Link prediction results on FB15k-237 vs FB15k

	FB15k-237						
Model	**MRR↑ (unidirectional)**	**MRR↑ (bidirectional)**	**MRR↑ (all)**	**MR↓**	**Hits@1↑**	**Hits@3↑**	**Hits@10↑**
TransE	0.358	0.227	0.245	257.750	0.146	0.281	0.442
DistMult	0.313	0.230	0.241	385.128	0.147	0.271	0.436
ComplEx	0.302	0.221	0.232	425.381	0.141	0.259	0.420
TransR	0.543	0.583	0.576	196.994	0.527	0.594	0.671
RotatE	0.396	0.224	0.246	288.433	0.160	0.267	0.424
	FB15k						
Model	**MRR↑(unidirectional)**	**MRR↑(bidirectional)**	**MRR↑(all)**	**MR↓**	**Hits@1↑**	**Hits@3↑**	**Hits@10↑**
TransE	0.561	0.638	0.631	46.556	0.497	0.736	0.839
DistMult	0.605	0.695	0.686	59.926	0.576	0.769	0.869
ComplEx	0.596	0.764	0.748	66.374	0.664	0.813	0.883
TransR	0.636	0.667	0.664	66.090	0.579	0.722	0.804
RotatE	0.638	0.685	0.680	50.282	0.575	0.759	0.850

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.

4.2 Mediator Nodes

Knowledge graph link prediction in the literature is conducted on binary relations in most cases. When multiary relationships (i.e., CVT nodes in Freebase) are present, link prediction could become more challenging due to several reasons. First, CVT nodes are long-tail nodes with limited connectivity, which makes link prediction harder. Second, training and evaluation setups might need to be specifically aware of the existence of CVT nodes. For example, if triples are randomly split into training/test/validation sets, since each CVT node is connected to only a few entities, many CVT nodes may appear in the test set without being existent in the training set. Nevertheless, 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 Section 7 presents the full results, here we highlight the most important observations.

Figure 4 shows the performance (MRR↑) of various models on two of our new datasets, FB-CVT-REV and FB+CVT-REV (dataset details in Section 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 Section 3 and detailed further in Section 6. In Figure 4, the blue and red dashed polygons depict various models' performance on FB-CVT-REV and FB+CVT-REV, respectively. All models performed worse when CVT nodes are present, verifying our earlier analysis of challenges posed by CVT nodes.

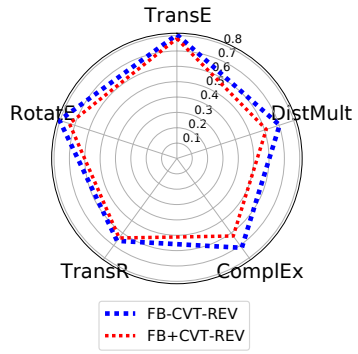


Figure 4: Impact of mediator nodes on the performance (MRR↑) of embedding models

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 as discussed in Section 3. In Table 3, the columns labeled "all" correspond to Figure 4 and are for both categories of relations together. The table has separate columns for binary and concatenated/multiary relations. These

Table 3: Link prediction performance (MRR↑) on FB-CVT-REV and FB+CVT-REV

	FB-CVT-REV			FB+CVT-REV		
Model	**binary**	**concatenated**	**all**	**binary**	**multiary**	**all**
TransE	0.758	0.970	0.806	0.780	0.986	0.781
DistMult	0.648	0.894	0.703	0.611	0.775	0.612
ComplEx	0.665	0.905	0.719	0.623	0.800	0.624
TransR	0.588	0.922	0.663	0.639	0.872	0.640
RotatE	0.768	0.925	0.804	0.735	0.889	0.736

results show that almost all 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.

It is worth noting that the row of TransE in Table 3 is a case of Simpson's paradox. More specifically, the model has worse performance on both concatenated (versus multiary in FB+CVT-REV) and binary relations in FB-CVT-REV, but its overall performance on FB-CVT-REV is better than FB+CVT-REV. This is because the models performed better on concatenated/multiary relations than binary ones and concatenated relations outnumber corresponding multiary relations—for example, 4 triples connected to the same CVT node become $\binom{4}{2} = 6$ triples after concatenation.

For link prediction on knowledge graphs containing multiary relationships, a few studies built models for data represented as hyper-relational facts [19, 48, 56], in which a multiary relationship is modeled as a set of key-value (relation-entity) pairs ($r_1 : e_1$, $r_2 : e_2$, ..., $r_n : e_n$), e.g., (*/award/award_nominations/award_nominee* : `James Ivory`, */award/award_nominations/award* : `BAFTA Award For Best Film`, */award/award_nominations/nominated_for* : `A Room With A View`). A similar but different representation [36] is to model a hyper-relational fact (s, p, o, Q) as a primary triple (s, p, o) coupled with a set of key-value pairs (qualifiers) Q. Note that there is a divide between these studies and the more conventional link prediction models such as TransE [7], ComplEx [43], and so on, in terms of both applicable datasets and methodologies. Conventional models cannot be applied on hyper-relational datasets, e.g., JF17K [48], because representation based on key-value pairs is alien to such models. Our work focuses on these conventional models. The datasets we create and use capture multiary relationships in Freebase through triples containing CVT nodes. In principle, the models built for hyper-relational facts could be applied on conventional datasets as well. We are unaware of any such empirical study, though, not to mention such studies on datasets containing CVT nodes. In fact, as discussed in Section 1, there does not exist a full-scale Freebase dataset that is properly prepared for tasks such as link prediction. In this regard, given the datasets and experiment results made available in this paper, it becomes possible to compare the performance of both hyper-relational fact models and conventional models on a full-scale Freebase dataset that includes multiary relationships (specifically, FB+CVT-REV in Table 6 of Section 6). Such a comparison will be the first not only for Freebase but also for any large-scale knowledge graph. For instance, to the best of our knowledge, there does not exist a full-scale Wikidata dataset with multiary relationships represented as conventional triples instead of hyper-relational facts. Therefore, conventional models have only been applied on Wikidata without multiary relationships [45], e.g., OGBL-WikiKG2 [23]. There exists no comparison of the two categories of models on Wikidata with multiary relationships.

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

4.3 Metadata and Administrative Data

As stated in [10], 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 [17], */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 class the subject of any triple that uses p as its predicate belongs, and *rdfs:range* denotes the type of the object of any such triple [3]. For example, the domain and range of the predicate */film/director/film* are *director* and *film*, respectively. Note that *rdfs:domain* and *rdfs:range* may appear to make part of the type system (Section 3) redundant, since they also establish the mapping from property to subject/object types. However, they have very limited coverage. More specifically, among the 7,881 properties belonging to subject matter domains in the original Freebase data dump, only 92 properties have *rdfs:domain* and only the same 92 properties have *rdfs:range*. On the other hand, the type system covers 2,891 out of 3,055 properties belonging to subject matter domains in our dataset FB-CVT-REV, as explained in Section 3.

Table 5: Link prediction performance (MRR↑) on Freebase86m and FB+CVT+REV

	Freebase86m			FB+CVT+REV
Model	**subject matter**	**non-subject matter**	**all**	**all**
TransE	0.746	0.686	0.729	0.970
DistMult	0.911	0.640	0.833	0.927
ComplEx	0.912	0.642	0.835	0.928
TransR	0.768	0.390	0.659	0.935
RotatE	0.927	0.567	0.823	0.948

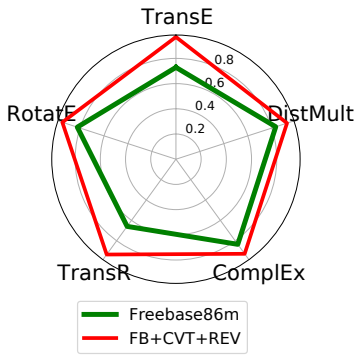


Figure 5: Impact of non-subject matter triples on the performance (MRR↑) of embedding models

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 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 above 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 Section 3. The details of this process are discussed in Section 6.

Figure 5 shows the impact of non-subject matter triples by comparing the performance (MRR↑) of link prediction models on Freebase86m (green polygon) and our new dataset FB+CVT+REV (red polygon), which includes only subject matter triples. The figure clearly illustrates 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.

5 OVERVIEW 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 [7] includes entities with at least 100 appearances in Freebase that were also available in Wikipedia based on the *wiki-links* database [11]. 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. Details of these issues were discussed thoroughly in [2].

FB15k-237 [42], 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, we observed that FB15k-237 includes both */education/educational_institution_campus/educational_institution* and */education/educational_institution/campuses* 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 [29, 57]. 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 on this dataset 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 Section 3 and 4, 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.

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

6 DATA PREPARATION

Variants of the Freebase Dataset We created four variants of the Freebase dataset by inclusion/exclusion of reverse triples and mediator (CVT) nodes. The datasets and data preprocessing scripts are made publicly available at https://github.com/idirlab/freebases. The variants allow one to examine the impact of the aforementioned Freebase idiosyncrasies on the effectiveness of knowledge graph completion methods. Beyond knowledge graph completion, these variants enable one to easily leverage or avoid the idiosyncrasies based on the nature of their task. 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. Hence, using these variants allows us to properly conduct link prediction experiments on subject matter triples only. 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.

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 Section 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 (`Rowan Atkinson`, *student*, `CVT`) and (`CVT`, *degree*, `Bachelor's degree`) in Figure 2a would be concatenated to form a new triple (`Rowan Atkinson`, *student–degree*, `Bachelor's degree`), as shown in Figure 2b. As briefly discussed in Section 3, in converting n-ary relationships to binary relationships, the concatenation does not need to be carried out along edges in the same direction.

7 EXPERIMENTS

Tasks Embedding models have been evaluated using several highly-related knowledge graph completion tasks such as triple classification [39, 47], link prediction [27], relation extraction [28, 49], and relation prediction [38]. The *link prediction* task as described in [7] 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 $rank_{\mathtt{h}}$. The same procedure is used to calculate $rank_{\mathtt{t}}$ 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 [7] and follow-up studies, including `Hits@1`$^{\uparrow}$, `Hits@3`$^{\uparrow}$, `Hits@10`$^{\uparrow}$, `MR`$^{\downarrow}$ (Mean Rank), and `MRR`$^{\uparrow}$ (Mean Reciprocal Rank). By definition as follows, higher `Hits@1`$^{\uparrow}$, `Hits@3`$^{\uparrow}$, `Hits@10`$^{\uparrow}$ and `MRR`$^{\uparrow}$, and lower `MR`$^{\downarrow}$ indicate better accuracy. An upward/downward arrow beside a measure indicates that methods with greater/smaller values by that measure possess higher accuracy. `Hits@k`$^{\uparrow}$ is the percentage of top k ranked triples that are correct. `MR`$^{\downarrow}$ is the mean of the test triples' ranks, defined as $MR = \frac{1}{2\,|T|}\sum_{(\mathtt{h},r,\mathtt{t})\in T}(rank_{\mathtt{h}} + rank_{\mathtt{t}})$ in which $|T|$ is the size of the test set. `MRR`$^{\uparrow}$ is the average inverse of harmonic mean of the test triples' ranks, defined as $MRR = \frac{1}{2\,|T|}\sum_{(\mathtt{h},r,\mathtt{t})\in T}(\frac{1}{rank_{\mathtt{h}}} + \frac{1}{rank_{\mathtt{t}}})$.

Instead of directly using the above-mentioned raw metrics', we use their corresponding *filtered* metrics [7], 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. For example, consider the task of predicting tail entity. Suppose the test

triple is (Tim Burton, *film*, Edward Scissorhands) and the training, test, or validation set also contains another triple (Tim Burton, *film*, Alice in Wonderland). If a model ranks Alice in Wonderland higher than Edward Scissorhands, the filtered metrics will remove this film from the ranked list so that the model would not be penalized for ranking (Tim Burton, *film*, Edward Scissorhands) lower than (Tim Burton, *film*, Alice in Wonderland), both correct triples.

Models We trained and evaluated five well-known link prediction embedding models—TransE [7], TransR [28], DistMult [54], ComplEx [43], and RotatE [41]—on the four variant datasets of Freebase discussed in Section 6. TransE, RotatE and TransR are three representative translational distance models. DistMult and ComplEx are semantic matching models that exploit similarity-based scoring functions [57].

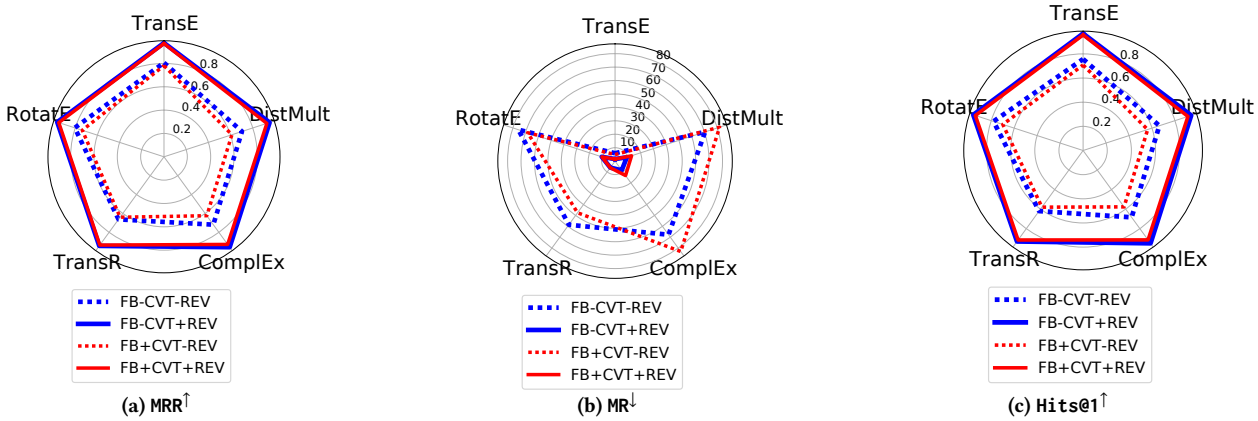
Experiment setup Multi-processing, multi-GPU distributed training frameworks have recently become available to scale up embedding models [26, 57, 58]. Our experiments were conducted using one such framework, DGL-KE [57] (https://github.com/awslabs/dgl-ke), with the settings and hyperparameters suggested in [57]. The experiments used an Intel-based machine with an Xeon E5-2695 processor running at 2.1GHz, 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 discussed in Section 4.2, blindly applying link prediction models when CVT nodes are present could be problematic. Hence, in the two datasets with CVT nodes, FB+CVT-REV and FB+CVT+REV, we made sure to split the data in such a way that a CVT node present in the test or validation set is also present in the training set.

Results on full-scale vs. small-scale Freebase datasets The results of our experiments are reported in Table 7 which is further visualized as radar charts in Figure 6. We also report results on Freebase86m in Table 7. 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 Section 5. 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 experiment results are in Table 2), respectively. Comparing the results on FB-CVT+REV and FB15k, in which reverse triples are retained, we can observe that models have much stronger performance on the full-scale dataset FB-CVT+REV. For example, LibKGE [8] (https://github.com/uma-pi1/kge) reported MRR$^{\uparrow}$ of TransE as 0.676 on FB15k, while our experiment results in Table 7 show 0.958 MRR$^{\uparrow}$ of TransE on FB-CVT+REV. Similarly, comparing the results on FB-CVT-REV and FB15k-237, both with reverse triples removed, the models again have substantially better accruacy when they are trained on the full-scale dataset FB-CVT-REV. For instance, the MRR$^{\uparrow}$ of ComplEx on FB15k-237 is reported by LibKGE as 0.348, which is considerably lower than the 0.717 obtained on FB-CVT-REV using DGL-KE. 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 noting 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.

Table 7: Link prediction performance on our four new variants of Freebase and Freebase86m

	FB-CVT-REV				
Model	**MRR$^{\uparrow}$**	**MR$^{\downarrow}$**	**Hits@1$^{\uparrow}$**	**Hits@3$^{\uparrow}$**	**Hits@10$^{\uparrow}$**
TransE	0.806	5.869	0.757	0.837	0.884
DistMult	0.703	70.498	0.664	0.724	0.775
ComplEx	0.719	67.740	0.684	0.738	0.783
TransR	0.663	58.553	0.620	0.684	0.743
RotatE	0.804	75.721	0.780	0.817	0.845
	FB-CVT+REV				
Model	**MRR$^{\uparrow}$**	**MR$^{\downarrow}$**	**Hits@1$^{\uparrow}$**	**Hits@3$^{\uparrow}$**	**Hits@10$^{\uparrow}$**
TransE	0.976	1.529	0.968	0.982	0.988
DistMult	0.952	9.239	0.941	0.960	0.970
ComplEx	0.958	8.437	0.950	0.964	0.972
TransR	0.944	5.982	0.931	0.952	0.967
RotatE	0.962	10.431	0.956	0.966	0.974
	FB+CVT-REV				
Model	**MRR$^{\uparrow}$**	**MR$^{\downarrow}$**	**Hits@1$^{\uparrow}$**	**Hits@3$^{\uparrow}$**	**Hits@10$^{\uparrow}$**
TransE	0.781	4.850	0.708	0.835	0.902
DistMult	0.612	81.841	0.562	0.635	0.704
ComplEx	0.624	83.205	0.577	0.647	0.708
TransR	0.640	47.524	0.580	0.669	0.754
RotatE	0.736	68.436	0.699	0.754	0.807
	FB+CVT+REV				
Model	**MRR$^{\uparrow}$**	**MR$^{\downarrow}$**	**Hits@1$^{\uparrow}$**	**Hits@3$^{\uparrow}$**	**Hits@10$^{\uparrow}$**
TransE	0.970	1.464	0.957	0.982	0.989
DistMult	0.927	12.924	0.913	0.935	0.951
ComplEx	0.928	13.278	0.915	0.935	0.951
TransR	0.935	6.071	0.916	0.948	0.969
RotatE	0.948	10.263	0.938	0.954	0.969
	Freebase86m				
Model	**MRR$^{\uparrow}$**	**MR$^{\downarrow}$**	**Hits@1$^{\uparrow}$**	**Hits@3$^{\uparrow}$**	**Hits@10$^{\uparrow}$**
TransE	0.729	23.274	0.654	0.775	0.872
DistMult	0.833	45.54	0.813	0.842	0.871
ComplEx	0.835	46.558	0.817	0.842	0.867
TransR	0.659	71.913	0.612	0.682	0.744
RotatE	0.823	65.46	0.810	0.826	0.849

Impact of reverse relations As discussed in Section 4.1, previous studies using small-scale datasets show substantial over-estimation of link prediction models' accuracy when reverse triples were included. The impact of reverse relations at the scale of the full Freebase dataset was never studied before. This paper thus fills the gap. As Figure 6 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 a similar observation. So do the results on the two variants with CVT nodes—FB+CVT-REV and FB+CVT+REV. In Figure 6, the solid polygons show the performance of link prediction models on the two Freebase variants in which reverse triples exist—FB-CVT+REV in blue solid polygons, and FB+CVT+REV in red solid polygons. Correspondingly, the dashed polygons are for model performance on the two variants in which reverse triples are removed—FB-CVT-REV in blue dashed polygons, and FB+CVT-REV in red dashed polygons. Comparing solid polygons with their corresponding dashed polygons, we can observe significant accuracy over-estimation across the board due to inclusion of reverse triples.



(a) MRR↑ (b) MR↓ (c) Hits@1↑

Figure 6: Link prediction performance on our four new variants of Freebase

Table 8: Triple classification results on FB15k-237

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

Impact of mediator nodes As articulated in Section 4.2, no prior work has studied the impact of mediator nodes on link prediction, regardless of dataset scale. Comparing the results on the two variants without reverse triples—FB-CVT-REV (mediator nodes excluded) and FB+CVT-REV (mediator nodes included)—shows that the existence of CVT nodes led to weaker model accuracy. Although the results on FB-CVT+REV and FB+CVT+REV are overestimations 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). In Figure 6, the red polygons show the performance of link prediction models on the two Freebase variants in which CVT nodes exist—FB+CVT+REV in red solid polygons, and FB+CVT-REV in red dashed polygons. Correspondingly, the blue polygons are for model performance on the two variants in which CVT nodes are removed—FB-CVT+REV in blue solid polygons, and FB-CVT-REV in blue dashed polygons. Comparing blue polygons with their corresponding red polygons, we can observe accuracy degeneration when CVT nodes are retained. Since the reverse triples lead to a significant overestimation in the overall performance of the models, this observation is more evident in the datasets where reverse triples are removed.

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 Section 4.2. Our newly created datasets will facilitate research in this direction.

Freebse86m vs. FB+CVT+REV As explained in Section 5, Freebase86m has both CVT nodes and reverse triples. FB+CVT+REV, one of the four Freebase variants we created, resembles Freebase86m since it retains these two Freebase idiosyncrasies as well. However, a notable distinction is that non-subject matter triples are removed from FB+CVT+REV while 31% of the triples in Freebase86m are non-subject matter triples, as mentioned in Section 5. Comparing the results on FB+CVT+REV and Freebse86m, as shown in Table 7 and visualized in Figure 5 for one of the measures, we observe 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.

Performance of models on different domains We further examined the performance of link prediction models separately on the 12 most frequent domains in FB-CVT-REV, which account for

Table 9: The most frequent subject-matter domains in FB-CVT-REV

Domain	#Triples	#Relations	TransR MRR↑	DistMult MRR↑	ComplEx MRR↑	TransE MRR↑	RotatE MRR↑
/music/	76,853,315	119	0.575	0.674	0.694	0.752	0.819
/film/	8,099,424	113	0.851	0.788	0.815	0.943	0.871
/people/	7,737,375	71	0.577	0.638	0.652	0.853	0.659
/tv/	5,276,787	148	0.926	0.922	0.929	0.970	0.945
/book/	5,258,889	90	0.531	0.284	0.268	0.590	0.290
/measurement_unit/	4,496,699	239	0.999	0.994	0.995	0.999	0.990
/location/	3,198,389	157	0.953	0.791	0.810	0.900	0.835
/award/	3,112,556	60	0.981	0.981	0.984	0.995	0.987
/biology/	1,459,238	71	0.573	0.596	0.606	0.814	0.560
/organization/	1,223,205	55	0.853	0.710	0.734	0.883	0.766
/education/	1,083,133	82	0.783	0.740	0.658	0.866	0.696
/sports/	996,040	105	0.967	0.871	0.902	0.973	0.910

almost 95% of all subject matter triples in the dataset. Table 9 reports, for each domain, the number of distinct relations, the number of triples, and the performance (MRR↑) of models. Overall, the results show that the models' accuracy vary a lot by domains. The overall performance of a model is heavily influenced by the dominating domains, especially *music* which accounts for more than 60% of the triples in FB-CVT-REV. This suggests that it is well justified to investigate and compare model performance on various datasets using macro average measures that give each domain equal importance, while existing studies only use micro average measures which give equal importance to each triple. To the best of our knowledge, this important matter has not been looked at in the literature. Hence, it is a major future direction for our study.

It is also important to understand the models' performance on a domain based on each individual domain's characteristics, given the large variance of performance across different domains. For instance, the models had almost perfect accuracy on domain *measurement_unit*. Most of the frequent relations in this domain are *1-to-n* relations. One example is */measurement_unit/dated_integer/source-/location/hud_foreclosure_area/estimated_number_foreclosures*, which is a concatenated property. Given the similarity between those objects, it could be easier to predict the subject. On the other hand, the models had particularly low accuracy on domains such as *book*. One of the most prevalent relations in this domain is */book/book_edition/isbn*. An ISBN value mostly participates in one triple only, and that triple would be an instance of */book/book_edition/isbn*. To predict either the book or the ISBN value in this instance, without the ISBN value being connected to anything else, the model would be helpless.

Usefulness of the type system To demonstrate the usefulness of the Freebase type system we created (Section 3), we evaluated embedding models' performance on the task of triple classification [47] using the LibKGE library [8]. 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 [39, 47], 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 Section 3, leading to overestimated classification accuracy. Pezeshkpour et al. [33] and Safavi et al. [37] 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. Based on these results, our immediate next step is to conduct similar experiments on our large-scale datasets.

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 and thus to facilitate robust development of knowledge graph completion technologies fully leveraging Freebase, 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.

REFERENCES

[1] Farahnaz Akrami, Lingbing Guo, Wei Hu, and Chengkai Li. 2018. Re-evaluating Embedding-Based Knowledge Graph Completion Methods. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. Association for Computing Machinery, Turin, Italy, 1779–1782. https://doi.org/10.1145/3269206.3269266

[2] Farahnaz Akrami, Mohammed Samiul Saeef, Qingheng Zhang, Wei Hu, and Chengkai Li. 2020. 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*. Association for Computing Machinery, Portland, Oregon, USA, 1995–2010. https://doi.org/10.1145/3318464.3380599

[3] Dean Allemang and James Hendler. 2011. *Semantic web for the working ontologist: effective modeling in RDFS and OWL*. Elsevier, Online.

[4] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. DBPedia: A nucleus for a web of open data. In *The semantic web*. Springer, Busan, Korea, 722–735.

[5] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. 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*. Association for Computing Machinery, Vancouver, Canada, 1247–1250.

[6] Kurt Bollacker, Patrick Tufts, Tomi Pierce, and Robert Cook. 2007. A platform for scalable, collaborative, structured information integration. In *Intl. Workshop on Information Integration on the Web*. AAAI Press, Vancouver, British Columbia, 22–27.

[7] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*. Curran Associates, Lake Tahoe, Nevada, United States, 2787–2795.

[8] Samuel Broscheit, Daniel Ruffinelli, Adrian Kochsiek, Patrick Betz, and Rainer Gemulla. 2020. LibKGE - A Knowledge Graph Embedding Library for Reproducible Research. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Online, 165–174. https://www.aclweb.org/anthology/2020.emnlp-demos.22

[9] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka, and Tom M Mitchell. 2010. Toward an architecture for never-ending language learning. In *Twenty-Fourth AAAI conference on artificial intelligence*. AAAI Press, New York, USA, 1306–1313.

[10] Niel Chah. 2017. Freebase-triples: A methodology for processing the freebase data dumps. *arXiv preprint* arXiv:1712.08707 (2017), arXiv–1712.

[11] Google Code. 2012. Wikipedia Links Data. https://code.google.com/archive/p/wiki-links/. Accessed: 2022-06-09.

[12] Tim Dettmers, Minervini Pasquale, Stenetorp Pontus, and Sebastian Riedel. 2018. Convolutional 2D Knowledge Graph Embeddings. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*. AAAI Press, New Orleans, Louisiana, USA, 1811–1818.

[13] Jason Douglas. 2015. Announcement: From Freebase to Wikidata. https://groups.google.com/g/freebase-discuss/c/s_BPoL92edc/m/Y585r7_2E1YJ. Accessed: 2015-02-17.

[14] Michael Färber. 2017. *Semantic Search for Novel Information*. IOS Press, Amsterdam, The Netherlands, The Netherlands.

[15] Michael Färber, Frederic Bartscherer, Carsten Menne, and Achim Rettinger. 2018. Linked data quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO. *Semantic Web* 9, 1 (2018), 77–129.

[16] David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A Kalyanpur, Adam Lally, J William Murdock, Eric Nyberg, John Prager, et al. 2010. Building Watson: An overview of the DeepQA project. *Artificial Intelligence magazine* 31, 3 (2010), 59–79.

[17] Google. 2013. Freebase Data Dumps. https://developers.google.com/freebase. Accessed: 2022-11-11.

[18] Jan Grant and Dave Beckett. 2004. RDF Test Cases. https://www.w3.org/TR/rdf-testcases/.

[19] Saiping Guan, Xiaolong Jin, Yuanzhuo Wang, and Xueqi Cheng. 2019. Link prediction on n-ary relational data. In *The World Wide Web Conference*. Association for Computing Machinery, San Francisco, CA, USA, 583–593.

[20] Shu Guo, Quan Wang, Bin Wang, Lihong Wang, and Li Guo. 2015. 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 (Volume 1: Long Papers)*. Association for Computational Linguistics, Beijing, China, 84–94. https://doi.org/10.3115/v1/P15-1009

[21] Yanchao Hao, Yuanzhe Zhang, Kang Liu, Shizhu He, Zhanyi Liu, Hua Wu, and Jun Zhao. 2017. 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 (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, 221–231.

[22] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems* 33 (2020), 22118–22133.

[23] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems* 33 (2020), 22118–22133.

[24] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and S Yu Philip. 2021. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems* 33, 2 (2021), 494–514.

[25] Graham Klyne. 2004. Resource description framework (RDF): Concepts and abstract syntax. http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/.

[26] Adam Lerer, Ledell Wu, Jiajun Shen, Timothee Lacroix, Luca Wehrstedt, Abhijit Bose, and Alex Peysakhovich. 2019. Pytorch-biggraph: A large scale graph embedding system. *Proceedings of Machine Learning and Systems* 1 (2019), 120–131.

[27] Yankai Lin, Zhiyuan Liu, Huanbo Luan, Maosong Sun, Siwei Rao, and Song Liu. 2015. Modeling Relation Paths for Representation Learning of Knowledge Bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, 705–714. https://doi.org/10.18653/v1/D15-1082

[28] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion.. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*. AAAI Press, Austin, Texas, USA, 2181–2187.

[29] Jason Mohoney, Roger Waleffe, Henry Xu, Theodoros Rekatsinas, and Shivaram Venkataraman. 2021. Marius: Learning Massive Graph Embeddings on a Single Machine. In *15th USENIX Symposium on Operating Systems Design and Implementation*. The Advanced Computing Systems Association, Online, 533–549.

[30] Networking, information technology research, and development. 2018. Open Knowledge Network: Summary of the Big Data IWG Workshop. https://www.nitrd.gov/open-knowledge-network-summary-of-the-big-data-iwg-workshop/.

[31] Natasha Noy, Yuqing Gao, Anshu Jain, Anant Narayanan, Alan Patterson, and Jamie Taylor. 2019. Industry-scale knowledge graphs: lessons and challenges. *Commun. ACM* 62, 8 (2019), 36–43.

[32] Thomas Pellissier Tanon, Denny Vrandečić, Sebastian Schaffert, Thomas Steiner, and Lydia Pintscher. 2016. From Freebase to Wikidata: The Great Migration. In *Proceedings of the 25th International Conference on World Wide Web*. Association for Computing Machinery, Montreal, Canada, 1419–1428.

[33] Pouya Pezeshkpour, Yifan Tian, and Sameer Singh. 2020. Revisiting Evaluation of Knowledge Base Completion Models. In *Automated Knowledge Base Construction*. OpenReview, Online, 10 pages. https://doi.org/10.24432/C53S3W

[34] Andrea Rossi, Denilson Barbosa, Donatella Firmani, Antonio Matinata, and Paolo Merialdo. 2021. Knowledge graph embedding for link prediction: A comparative analysis. *ACM Transactions on Knowledge Discovery from Data* 15, 2 (2021), 1–49.

[35] Andrea Rossi, Denilson Barbosa, Donatella Firmani, Antonio Matinata, and Paolo Merialdo. 2021. Knowledge graph embedding for link prediction: A comparative analysis. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 15, 2 (2021), 1–49.

[36] Paolo Rosso, Dingqi Yang, and Philippe Cudré-Mauroux. 2020. Beyond triplets: hyper-relational knowledge graph embedding for link prediction. In *Proceedings of The Web Conference 2020*. Association for Computing Machinery, Online, 1885–1896.

[37] Tara Safavi and Danai Koutra. 2020. CoDEx: A Comprehensive Knowledge Graph Completion Benchmark. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online, 8328–8350. https://doi.org/10.18653/v1/2020.emnlp-main.669

[38] Baoxu Shi and Tim Weninger. 2017. ProjE: Embedding projection for knowledge graph completion. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*. AAAI Press, San Francisco, California, USA, 1236–1242.

[39] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*. Curran Associates, Lake Tahoe, United States, 926–934.

[40] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2008. Yago: A large ontology from wikipedia and wordnet. *Journal of Web Semantics* 6, 3 (2008), 203–217.

[41] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *Proceedings of the International Conference on Learning Representations*. OpenReview.net, New Orleans, LA, USA, 926–934.

[42] Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*. Association for Computational Linguistics, Beijing, China, 57–66. https://doi.org/10.18653/v1/W15-4007

[43] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of the 33rd International Conference on Machine Learning*. JMLR.org, New York City, NY, USA, 2071–2080.

[44] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85.

[45] Huijuan Wang, Siming Dai, Weiyue Su, Hui Zhong, Zeyang Fang, Zhengjie Huang, Shikun Feng, Zeyu Chen, Yu Sun, and Dianhai Yu. 2022. Simple and Effective Relation-based Embedding Propagation for Knowledge Representation Learning. *International Joint Conference on Artificial Intelligence (IJCAI)* arXiv preprint (2022), arXiv:2205.06456.

[46] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering* 29, 12 (2017), 2724–2743.

[47] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge Graph Embedding by Translating on Hyperplanes. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*. AAAI Press, Québec City, Québec, Canada, 1112–1119.

[48] Jianfeng Wen, Jianxin Li, Yongyi Mao, Shini Chen, and Richong Zhang. 2016. On the Representation and Embedding of Knowledge Bases beyond Binary Relations. In *Proceedings of the International Joint Conference on Artificial Intelligence*. IJCAI, New York City, USA, 1300–1307.

[49] Jason Weston, Antoine Bordes, Oksana Yakhnenko, and Nicolas Usunier. 2013. Connecting Language and Knowledge Bases with Embedding Models for Relation Extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Seattle, USA, 1366–1371.

[50] Wikidata. 2015. Wikidata:WikiProject Freebase. https://www.wikidata.org/wiki/Wikidata:WikiProject_Freebase. Accessed: 2022-06-09.

[51] Ruobing Xie, Zhiyuan Liu, Maosong Sun, et al. 2016. Representation Learning of Knowledge Graphs with Hierarchical Types.. In *Proceedings of International Joint Conference on Artificial Intelligence*, Vol. 2016. IJCAI, New York City, USA, 2965–2971.

[52] Chenyan Xiong, Russell Power, and Jamie Callan. 2017. Explicit semantic ranking for academic search via knowledge graph embedding. In *Proceedings of the 26th international conference on world wide web*. Association for Computing Machinery, Perth, Australia, 1271–1279.

[53] Bishan Yang and Tom Mitchell. 2017. Leveraging Knowledge Bases in LSTMs for Improving Machine Reading. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, 1436–1446.

[54] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of the International Conference on Learning Representations*. OpenReview.net, San Diego, CA, USA, 12.

[55] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. Association for Computing Machinery, San Francisco, CA, USA, 353–362.

[56] Richong Zhang, Junpeng Li, Jiajie Mei, and Yongyi Mao. 2018. Scalable instance reconstruction in knowledge bases via relatedness affiliated embedding. In *Proceedings of the 2018 World Wide Web Conference*. Association for Computing Machinery, Lyon, France, 1185–1194.

[57] Da Zheng, Xiang Song, Chao Ma, Zeyuan Tan, Zihao Ye, Jin Dong, Hao Xiong, Zheng Zhang, and George Karypis. 2020. DGL-KE: Training Knowledge Graph Embeddings at Scale. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, Xi'an, China, 739–748.

[58] Zhaocheng Zhu, Shizhen Xu, Meng Qu, and Jian Tang. 2019. GraphVite: A High-Performance CPU-GPU Hybrid System for Node Embedding. In *The World Wide Web Conference*. Association for Computing Machinery, San Francisco, CA, USA, 2494–2504.