Creating Variants of Freebase for Robust Development of Intelligent Tasks on Knowledge Graphs

Farahnaz Akrami*, **Mohammed Samiul Saeef***, **Nasim Shirvani-Mahdavi***, **Xiao Shi, Chengkai Li**
Department of Computer Science and Engineering, University of Texas at Arlington
{farahnaz.akrami, mohammedsamiul.saeef,
nasim.shirvanimahdavi2, xiao.shi}@mavs.uta.edu
cli@uta.edu

Abstract

Knowledge graphs (KGs) are an essential asset to a wide variety of tasks and applications. They encode rich semantic, factual information, and different datasets could be potentially linked together for purposes greater than what they support separately. Freebase is amongst the largest public cross-domain KGs that store common facts. It possesses several data modeling idiosyncrasies rarely found in comparable datasets. It has a strong type system; its properties are purposefully represented in reverse pairs; and it uses mediator objects to facilitate representation of multiary relationships. These design choices serve important practical purposes in realistically modeling the real-world. But they also pose nontrivial challenges that could hinder the advancement of KG-oriented technologies. More 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. This paper lays out a comprehensive analysis of the challenges associated with the aforementioned idiosyncrasies of Freebase, measures their impact on tasks such as link prediction, and provides several variants of the Freebase dataset by inclusion/exclusion of various data modeling idiosyncrasies. The datasets and data preparation scripts are publicly available. They can be a valuable resource to researchers and practitioners in developing technologies by and for knowledge graphs.

1 Introduction

The ability to exploit big data on the Web enables intelligent systems [38]. Such data include encyclopedic knowledge of real-world factual information. Knowledge graphs (KGs) encode such semantic, factual information as triples of the form (subject, predicate, object). 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 [13, 25], including natural language processing [48], information retrieval and web search [47], knowledge-base question answering [22], and recommender systems [50]. Consequently, KGs are of great importance to many technology companies [29, 18] and governments [3].

To develop and robustly evaluate models and algorithms for intelligent tasks on knowledge graphs, access to large-scale KGs is crucial. But publicly available KG datasets are often much smaller than

*Equal Contribution

Submitted to the 36th Conference on Neural Information Processing Systems (NeurIPS 2022) Track on Datasets and Benchmarks. Do not distribute.

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 knowledge graphs are both large and publicly available, e.g., DBpedia [7], Freebase [8], Wikidata [41], YAGO [36], 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, making it relevant to many applications. The dataset possesses several data modeling idiosyncrasies rarely found in the aforementioned comparable datasets. These design choices serve important practical purposes in realistically 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 [30]. *Secondly*, Freebase uses mediator objects to facilitate representation of n-ary relationships [30]. *Lastly*, Freebase's strong type system categorizes each entity into one or more types, and the type of an entity determines the properties it may possess [9]. Furthermore, in practice the label of a property *almost* functionally determines the types of the entities in its two ends. As a simple example of the type system's merits, when querying the graph, a filtering condition on entities can be specified using an entity type.

Albeit highly useful, the aforementioned idiosyncrasies also pose nontrivial challenges that could hinder the advancement of KG-oriented technologies. More 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 many knowledge graph link prediction models [43, 32] 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 evaluating the models. The consequence is substantial over-estimation of the models' accuracy and unrealistic comparison of their relative strengths [5].

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 provides four variants of the Freebase dataset by inclusion/exclusion of mediator objects and reverse triples. A Freebase type system is extracted and included in all these variants. Furthermore, the datasets underwent thorough cleaning in order to improve their utility and to remove irrelevant triples from the original Freebase data dump.

The methodology, code, datasets, and experiment results produced from this work 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 data modeling idiosyncrasies such as mediator objects, reverse triples, and type system, 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; Freebase86m [52], one of the few available full-scale Freebase datasets, also mixes together everything—operational data that are not common knowledge facts, reverse triples, and mediator objects. (Details in Section 5.)

The paper also fills an important gap in our understanding of knowledge graph embedding models. 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 the aforementioned unreliable results; and 2) how data idiosyncrasies such as mediator objects and reverse triples impact model performance on the complete Freebase data.

The datasets and results are highly relevant to the research community, 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 63 publications. [2] Among them, 57 publications used datasets produced from Freebase, while 9 used datasets from Wikidata. Only 2 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 the lack of proper datasets.

The dataset creation was nontrivial and time-consuming. 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 researchers must learn to examine Freebase and prepare datasets from scratch, we expect many of them to have steep learning curve and the process can easily require months or even years. Our datasets can thus speed up many researchers' work.

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 performance of conventional embedding models (e.g., TransE [10] and ComplEx [40]) and hyper-relational fact models [45, 51, 20, 33] on a full-scale Freebase dataset that includes multiary relationships (i.e., mediator objects). 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 4.1.

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 [9, 26, 19, 30] in our description.

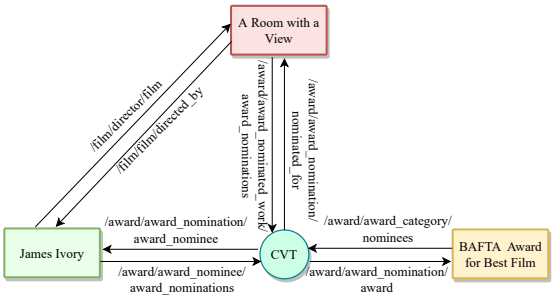

Figure 1: An example of a mediator node in Freebase

RDF: Freebase is stored in N-Triples RDF (Resource Description Format) [26]. 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): Freebase objects 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. In viewing Freebase as a graph, these topics correspond to nodes in the graph. However, not every node in the Freebase graph is a topic. CVT (Compound Value Type) nodes are examples of non-topic objects used to represent n-ary relations (details in Section 4.1). 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`.

Type: Types are used to group topics 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].

[2] 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`.

The /[domain]/[type] prefix helps explain the type of the topic 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. It also has another property */film/director/film*, on which the value is another topic `A Room with a View`, as shown in Figure 1. When a relationship is represented as a triple, the predicate in the triple is a property of the 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 or *edge type* can be uniquely identified by the label of the edge. 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 the schema of that type 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 KGs 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 matters domains*—domains describing real-world facts—ranging from American music to visual art [13]. As stated in [17], Freebase's data is consistent, correct, reliable, 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 [15]. This transfer is yet to be completed [30, 1]. Nonetheless, Freebase has several idiosyncrasies of data modeling design choices, which are rarely found in Wikidata and other comparable datasets, as follows.

Mediator Nodes *Mediator nodes*, also called CVT nodes, are used in Freebase to represent n-ary relationships [30]. 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. To reduce complexity of algorithmic solutions, one may convert an n-ary relationship centered at a CVT node into $\binom{n}{2}$ binary relationships between every pair of entities connected to the CVT node. Note that such a transformation leads to loss of information [45]. For instance, after such a transformation for Figure 1, the new triples cannot exactly pinpoint to the work that leads to `James Ivory`'s nomination for the BAFTA award.

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* [30, 16]. For instance, the triple (`/film/film/directed_by`, */type/property/reverse_property*, `/film/director/film`) denotes that */film/film/directed_by* and */film/director/film* are reverse relations. (`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, as Figure 1 shows. Reverse relations help traverse the graph in both directions [30].

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) having at least one property belonging to a type that is not among its 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. Out of 2,891

edge types (103,324,039 triples, again, after cleaning the data dump), 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 are interested in creating an explicit type system, which can become useful when enforced in various intelligent tasks. Note that Freebase itself does not explicitly specify such a type system, even though the data appear to be included while following guidelines that approximately form the type system, e.g., the "expected type" mentioned earlier. The goal 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*, *music/artist*, *award/award_winner*, *people/person*, *tv/tv_actor*}, 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*$) = 0.9969$, $P(o \in$ *music/artist*$) = 0.0477$, $P(o \in$ *award/award_winner*$) = 0.0373$, $P(o \in$ *people/person*$) = 0.998$, and $P(o \in$ *tv/tv_actor*$) = 0.1052$. Using $\alpha = 0.95$, *music/artist*, *award/award_winner* and *tv/tv_actor* were excluded. In *step 2*, we pick the most specific type among the remaining candidates. We use $P(n \in t_1 | n \in t_2)$ for the probability of a Freebase entity n belonging to type t_1 given that it also belongs to type t_2. In the dataset $P(n \in$ *people/person*$|n \in$ *film/actor*$) = 0.9984$ and $P(n \in$ *film/actor*$|n \in$ *people/person*$) = 0.1394$. Thus, we assigned *film/actor* as the required entity type for the object node of edge type *film/film/performance*.

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 a 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 $P(n \in t)$ as defined above. The average type assignment accuracy scores for node instances (both subject and object nodes) of all edge types we found to be 0.93707, 0.99134, 0.99413, and 0.99692 for $\alpha = 0.5$, $\alpha = 0.85$, $\alpha = 0.9$, and $\alpha = 0.95$ respectively. It shows that we are getting the highest accuracy for the chosen threshold. We also observed the impact of different values of α on the type system for a specific edge type. For 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. As observed, the assigned type becomes less specific as we increase α. We get the most appropriate type for the object node at $\alpha = 0.95$.

The type system we created can be useful in improving link prediction performance. A few studies employed type information for such a goal [46, 21]. Particularly, embedding models can aim to keep entities of the same type stay close to each other in the embedding space [21]. Further, type information could be a simple, effective feature of a model or used as a constraint while generating negative training or test examples. 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; on the other hand, 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 of Directly Using Freebase Data Dump

The latest Freebase data dump constitutes a snapshot of the data and its schema. As discussed earlier, the data modeling idiosyncrasies of Freebase could pose challenges that hamper the advancement of KG-oriented technologies. Specifically, when algorithms and models for intelligent tasks are developed and evaluated agnostically of these characteristics, they may fall into pitfalls without knowing. This section explains such challenges and their impact on tasks such as link prediction.

4.1 Mediator Nodes

Link prediction [10] in the literature is conducted on binary relations in most cases. When multiary relationships (e.g., CVT nodes) are present, link prediction could become more challenging due to several reasons. First, CVT nodes are long-tail nodes with limited structural information, 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 (details in Section 7).

For link prediction on knowledge graphs containing multiary relationships, a few studies built models for data represented as hyper-relational facts [45, 51, 20], 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 [33] 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 [10], ComplEx [40], and so on, in terms of both the applicable datasets and the methodologies. Conventional models cannot be applied on hyper-relational datasets, e.g., JF17K [45], 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, although we are unaware of any such empirical study, 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, FB3 in Table 1 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 [42], e.g., OGBL-WikiKG2 [24]. There exists no comparison of the two categories of models on Wikidata with multiary relationships.

4.2 Reverse Triples

Several previous studies discussed the problems of including the reverse relations in datasets used for link prediction [14, 39, 4, 5]. 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 subset of Freebase), created by Bordes et al. [10], is almost always used for knowledge graph link prediction. 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 idiosyncrasies of the link prediction task on datasets such as FB15K with many reverse triples 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 link prediction's 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 (`s`, p_1, `o`) $\Rightarrow$ (`o`, p_2, `s`) [14], 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

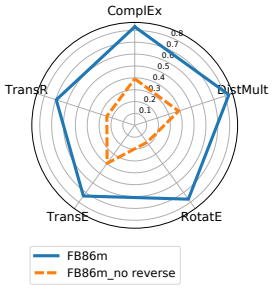

Figure 2: Performance (MRR$^{\uparrow}$) decrease of link prediction after removal of reverse triples

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* [30, 16]. 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.

The dataset FB1 presented in Section 7 is the first large-scale Freebase dataset without reverse triples. More than 38% of the triples in Freebase86m form reverse pairs. As mentioned earlier, including these triples will lead to overestimation of link prediction results. Figure 2 shows that link prediction models' accuracy degenerates drastically after removal of reverse triples from Freebase86m. Similar results were observed on FB15k vs. FB15k-237 [5]. The process of removing reverse relations is detailed in Section 6.

4.3 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. */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 [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 real-world facts. We call (s, *p*, o) a subject matter triple if the domains of s, *p* and o belong to subject matter. Computational tasks and applications thus need to be applied on this category of domains instead of the other two categories. However, in Freebase86m, OWL domains are removed but implementation domains are kept. About 31% of the Freebase86m triples fall under non-subject matter domains.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 in all datasets. The details of this process are discussed in Section 6.

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 is a subset of Freebase generated by Bordes et al. [10] for the task of *link prediction*. FB15k retains entities with at least 100 appearances in Freebase that were also available in Wikipedia based on the *wiki-links* database [2]. 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. The details of these issues are discussed thoroughly in [5].

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, due to two types of

mistakes. 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 latest Freebase data dump and is employed in evaluating multiple large-scale knowledge graph embedding frameworks [52, 28]. It includes 86,054,151 entities, 14,824 relations and 338,586,276 triples. No information is available on how this dataset was created from Freebase. 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 could be helpful for some tasks but also lead to substantial over-estimation of link predication models' accuracy; and the existence of mediator nodes presents extra challenges to models. Mixing all these triples together, without clear annotation and separation, leads to foreseeably unreliable models and results.

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. These variants allow one to easily leverage or avoid these idiosyncrasies based on the nature of their task. In all variants, the type system is included and metadata and administrative triples are detached from subject matter triples. Table 1 presents the statistics of these variants. The datasets and data preprocessing scripts are made publicly available at `https://github.com/idirlab/freebases`. The rest of this section provides some details about how these variants were created from the original Freebase data dump, which is nontrivial largely due to the scarcity of available documentation.

Table 1: Statistics of the four variants of Freebase

variant	CVT nodes	reverse triples	#entities	#relations	#triples
FB1	removed	removed	39,732,008	2,891	103,324,039
FB2	removed	retained	39,745,618	4,894	235,307,422
FB3	retained	removed	59,894,890	2,641	134,213,735
FB4	retained	retained	59,896,902	4,425	244,112,599

URI Simplification As mentioned in Section 2, Freebase is stored in N-Triples RDF format. Each component of a triple (`subject`, *predicate*, `object`), if not a literal, can be identified by a URI (uniform resource identifier) [26]. 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 that correspond to domains, types, properties' labels, and MIDs. These segments are dot-delimited in the URI. To make it more readable, 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 object labels are included in our dataset.

Extracting Metadata The non-subject matter triples are used to extract metadata about the subject matter triples. We created a mapping between Freebase objects (not to be confused with RDF objects) and their types using Freebase predicate */type/object/types*. Using predicate */type/object/name*, we created a lookup table mapping the MIDs of Freebase $entities$ 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. For instance, a triple (`r1`, */type/property/reverse_property*, `r2`) indicates that relations *r1* and *r2* are reverse of each other. When we remove reverse triples, 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 (see below). 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_category/nominees*, `CVT`) and (`CVT`, *award_nomination/nominated_for*, `A Room With A View`) in Figure 1 would be concatenated to form a new triple (`BAFTA Award For Best Film`, *award_category/nominees–award_nomination/nominated_for*, `A Room With A View`).

7 Experiments

We trained and evaluated link prediction embedding models on the aforementioned four variants of Freebase using TransE [10], DistMult [49], ComplEx [40], and RotatE [37]. The results are reported in Table 2. We measured their performance with commonly used metrics `MRR`$^{\uparrow}$(Mean Reciprocal Rank), `MR`$^{\downarrow}$(Mean Rank), `Hits@1`$^{\uparrow}$, and `Hits@10`$^{\uparrow}$(denoted `H1`$^{\uparrow}$and `H10`$^{\uparrow}$in the reesult tables to save space) [10]. All reported results use the filtered measures described in [10]. An upward/downward arrow beside a measure indicates that methods with greater/smaller values by that measure possess higher accuracy. Recently, multi-processing multi-GPU distributed training frameworks have become available in order to scale up embedding models [27, 53, 52]. Our experiments were conducted using one such framework, DGL-KE [52], with the same hyperparameters and settings suggested in [52]. 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 4.1, blindly applying link prediction models when CVT nodes are present could be problematic. Hence, in the two datasets with CVT nodes, FB3 and FB4, 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 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 FB1 and FB2 can be viewed as the full-scale counterparts of FB15k-237k and FB15k, respectively. Comparing the results on FB2 and FB15k, in which reverse triples are retained, we can observe that models have better performance on full-scale dataset, for example, LibKGE [11] (`https://github.com/uma-pi1/kge`) reports `MRR`$^{\uparrow}$of TransE as 0.676 while based on results in Table 2, its `MRR`$^{\uparrow}$is 0.958 on FB2. Similarly, comparing the results on FB1 and FB15k-237, both with reverse triples removed, the models again have better accruacy when they are trained on the full-scale dataset. 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 FB1 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 something worth knowing and not reported before. This accuracy difference could be attributed to the 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 As discussed in Section 4.2, previous studies show substantial overestimation of link prediction models' accuracy when reverse triples were included. Results on the two variants without CVT nodes—FB1 (excluded reverse relations) and FB2 (included reverse relations)—present a similar observation. So do the results on the two variants with CVT nodes, FB3 and FB4. The new contribution is that such an investigation at the scale of the full Freebase dataset was not reported before.

Impact of mediator nodes As articulated in Section 4.1, no prior work has studied the impact of mediator nodes on link prediction. Comparing the results on the two variants without reverse triples—FB1 (excluded mediator nodes) and FB3 (included mediator nodes)—shows that the existence of

Table 2: Link prediction results on FB1vs FB2vs FB3vs FB4

	FB1				FB2				FB3				FB4			
Model	MRR$^{\uparrow}$	MR$^{\downarrow}$	H1$^{\uparrow}$	H10$^{\uparrow}$	MRR$^{\uparrow}$	MR$^{\downarrow}$	H1$^{\uparrow}$	H10$^{\uparrow}$	MRR$^{\uparrow}$	MR$^{\downarrow}$	H1$^{\uparrow}$	H10$^{\uparrow}$	MRR$^{\uparrow}$	MR$^{\downarrow}$	H1$^{\uparrow}$	H10$^{\uparrow}$
TransE	0.686	41.810	0.625	0.799	0.958	3.857	0.944	0.980	0.431	50.572	0.339	0.623	0.606	12.542	0.515	0.771
DistMult	0.709	69.388	0.670	0.780	0.965	6.059	0.956	0.979	0.408	109.193	0.318	0.581	0.818	19.180	0.777	0.890
ComplEx	0.717	68.798	0.681	0.783	0.970	5.567	0.964	0.981	0.510	104.317	0.439	0.635	0.899	16.937	0.880	0.935
RotatE	0.455	143.688	0.399	0.559	0.938	13.513	0.926	0.956	0.198	195.001	0.147	0.292	0.729	33.027	0.683	0.816

Table 3: 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
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
RotatE	0.60	0.47	0.58	0.53	0.87	0.78	0.83	0.82

CVT nodes led to much weaker model accuracy. Although the results on FB2 and FB4 are overestimations since they both retained reverse triples, similar observation regarding mediator nodes is still made—the models are much less accurate on FB4 (included mediator nodes) than FB2 (excluded mediator nodes). More detailed analyses remain to be done, in order to break down the different impacts of individual factors that contribute to the performance degeneration, such as the factors analyzed in Section 4.1. Our newly created datasets will facilitate research in this direction.

Usefulness of the type system To show the usefulness of the Freebase type system created for our datasets, we evaluated embedding models' performance on the task of triple classification [44] using the LibKGE library [11]. This task is the binary classification of triples regarding whether they are true or false facts. We need to generate a set of negative triples to conduct this task. The type system proves useful in generating type-consistent negative samples. When triple classification was initially used for evaluating models [35, 44], 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, leading to overestimated classification accuracy. Pezeshkpour et al. [31] and Safavi et al. [34] 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 same type as t. 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 all these new test sets are presented in Table 3. We can observe 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 (CVT nodes, reverse properties, and type system). To tackle these challenges and thus to facilitate scalable and robust development of AI and machine learning technologies fully leveraging Freebase, we provide four variants of the Freebase dataset by inclusion/exclusion of these idiosyncrasies. Therefore, one can grasp the opportunity to leverage or avoid such features based on the nature of their tasks. Furthermore, the datasets underwent thorough cleaning in order to improve their utility and data quality. In all these variants, the Freebase type system is included. This is the first time that a group of datasets are created by carefully considering Freebase idiosyncrasies. We believe they can be a valuable resource to researchers and practitioners in developing technologies by and for knowledge graphs.

References

[1] Wikidata:Wikiproject Freebase. https://www.wikidata.org/wiki/Wikidata:WikiProject_Freebase. Accessed: 2022-06-09.

[2] Wikipedia links data. https://code.google.com/archive/p/wiki-links/. Accessed: 2022-06-09.

[3] Open knowledge network: Summary of the big data IWG workshop. https://www.nitrd.gov/open-knowledge-network-summary-of-the-big-data-iwg-workshop/, 2018.

[4] Farahnaz Akrami, Lingbing Guo, Wei Hu, and Chengkai Li. Re-evaluating embedding-based knowledge graph completion methods. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1779–1782, 2018. doi:10.1145/3269206.3269266.

[5] Farahnaz Akrami, Mohammed Samiul Saeef, Qingheng Zhang, Wei Hu, and Chengkai Li. 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*, page 1995–2010, 2020. doi:10.1145/3318464.3380599.

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

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

[8] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 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*, pages 1247–1250, 2008.

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

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

[11] Samuel Broscheit, Daniel Ruffinelli, Adrian Kochsiek, Patrick Betz, and Rainer Gemulla. LibKGE - A knowledge graph embedding library for reproducible research. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 165–174, 2020. URL: https://www.aclweb.org/anthology/2020.emnlp-demos.22.

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

[13] Niel Chah. Freebase-triples: A methodology for processing the freebase data dumps. *arXiv preprint arXiv:1712.08707*, 2017.

[14] Tim Dettmers, Minervini Pasquale, Stenetorp Pontus, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pages 1811–1818, 2018.

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

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

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

[18] 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. Building Watson: An overview of the DeepQA project. *Artificial Intelligence magazine*, 31(3):59–79, 2010.

[19] Jan Grant and Dave Beckett. RDF test cases. 2004. URL: `https://www.w3.org/TR/rdf-testcases/`.

[20] Saiping Guan, Xiaolong Jin, Yuanzhuo Wang, and Xueqi Cheng. Link prediction on n-ary relational data. In *The World Wide Web Conference*, pages 583–593, 2019.

[21] Shu Guo, Quan Wang, Bin Wang, Lihong Wang, and Li Guo. 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)*, pages 84–94. Association for Computational Linguistics, 2015. URL: `https://aclanthology.org/P15-1009`, `doi:10.3115/v1/P15-1009`.

[22] Yanchao Hao, Yuanzhe Zhang, Kang Liu, Shizhu He, Zhanyi Liu, Hua Wu, and Jun Zhao. 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)*, pages 221–231, 2017.

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

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

[25] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and S Yu Philip. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

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

[27] Adam Lerer, Ledell Wu, Jiajun Shen, Timothee Lacroix, Luca Wehrstedt, Abhijit Bose, and Alex Peysakhovich. PyTorch-BigGraph: A Large-scale Graph Embedding System. In *Proceedings of the 2nd SysML Conference*, Palo Alto, CA, USA, 2019.

[28] Jason Mohoney, Roger Waleffe, Henry Xu, Theodoros Rekatsinas, and Shivaram Venkataraman. Marius: Learning massive graph embeddings on a single machine. In *15th USENIX Symposium on Operating Systems Design and Implementation*, pages 533–549, 2021.

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

[30] Thomas Pellissier Tanon, Denny Vrandečić, Sebastian Schaffert, Thomas Steiner, and Lydia Pintscher. From Freebase to Wikidata: The great migration. In *Proceedings of the 25th International Conference on World Wide Web*, pages 1419–1428, 2016.

[31] Pouya Pezeshkpour, Yifan Tian, and Sameer Singh. Revisiting evaluation of knowledge base completion models. In *Automated Knowledge Base Construction*, 2020.

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

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

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

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

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

[37] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. In *Proceedings of the International Conference on Learning Representations*, pages 926–934, 2019.

[38] Niket Tandon, Aparna S Varde, and Gerard de Melo. Commonsense knowledge in machine intelligence. *The ACM Special Interest Group on Management of Data Record*, 46(4):49–52, 2018.

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

[40] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 2071–2080, 2016.

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

[42] Huijuan Wang, Siming Dai, Weiyue Su, Hui Zhong, Zeyang Fang, Zhengjie Huang, Shikun Feng, Zeyu Chen, Yu Sun, and Dianhai Yu. Simple and effective relation-based embedding propagation for knowledge representation learning. *International Joint Conference on Artificial Intelligence (IJCAI)*, 2022.

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

[44] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pages 1112–1119, 2014.

[45] Jianfeng Wen, Jianxin Li, Yongyi Mao, Shini Chen, and Richong Zhang. On the representation and embedding of knowledge bases beyond binary relations. In *Proceedings of the International Joint Conference on Artificial Intelligence*, page 1300–1307, 2016.

[46] Ruobing Xie, Zhiyuan Liu, Maosong Sun, et al. Representation learning of knowledge graphs with hierarchical types. In *Proceedings of International Joint Conference on Artificial Intelligence*, volume 2016, pages 2965–2971, 2016.

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

[48] Bishan Yang and Tom Mitchell. Leveraging knowledge bases in lstms for improving machine reading. *arXiv preprint arXiv:1902.09091*, 2019.

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

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

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

[52] Da Zheng, Xiang Song, Chao Ma, Zeyuan Tan, Zihao Ye, Jin Dong, Hao Xiong, Zheng Zhang, and George Karypis. DGL-KE: Training knowledge graph embeddings at scale. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 739–748. Association for Computing Machinery, 2020.

[53] Zhaocheng Zhu, Shizhen Xu, Meng Qu, and Jian Tang. Graphvite: A high-performance cpu-gpu hybrid system for node embedding. In *The World Wide Web Conference*, pages 2494–2504. ACM, 2019.

A Key Information

A.1 Hosting, licensing, and maintenance plan

We used the latest Freebase data dump available at `https://developers.google.com/freebase` to generate our datasets. Freebase data dump is distributed under the Creative Commons Attribution (aka CC-BY). Our datasets along with the scripts required to generate them from Freebase datadump are available in our GitHub repository `https://github.com/idirlab/freebases`, licensed under the CC-0 license. The Innovative Data Intelligence Research Lab at UTA, where the authors are, is committed to maintaining the datasets. The lab has a track record of maintaining multiple research prototypes, demos, and datasets at `https://idir.uta.edu/`. There are several projects underway in our lab using these datasets and we will update our GitHub repository with the latest results from these projects. Any further updates to the datasets will be posted there too. Furthermore, we plan to archive the datasets at Zenodo upon paper acceptance.

A.2 Intended uses

Our datasets are intended to be used by researchers and practitioners in developing technologies based on and for knowledge graphs.

A.3 Limitations

Freebase was shut down in 2015 and its latest data dump (used by us to generate our datasets) was made available in Aug of 2015. Hence, when using the datasets one should be aware of the probability of some triples being outdated. It is also worth mentioning that Freebase domains are not equally distributed. Although Freebase covers a diverse range of domains, not all of them are well represented. For example, domain */music/* constitutes around 6.7% of the triples while domain */tennis/* accounts for only 0.001% [13].

A.4 Potential negative impacts

As it was mentioned, Freebase domains are distributed unevenly. Distribution of entities and relations is also skewed, with some entities and relations being more popular than others. If a model is trained on our datasets and is used for some real-world tasks, the bias available in data may be present in the model's output as well.

A.5 Author statement

We bear all responsibility in case of violation of rights and confirm CC-0 licenses for the included datasets.

B Details of datasets, dataset format, and dataset creation scripts

The datasets and data preprocessing scripts are made publicly available at `https://github.com/idirlab/freebases`.

The data is stored in CSV files in the form of triples which is a widely used data format for storing knowledge graph data.

As discussed in Section 6 of the paper, we provide four variants of the Freebase dataset by inclusion/exclusion of some of the Freebase's idiosyncrasies. For each of these datasets, we made three kinds of files available:

- Metadata files:
  - `object_types`: Each row maps the MID of a Freebase object to a type it belongs to.

  - `object_ids`: Each row maps the MID of a Freebase object to its user-friendly identifier.
  - `object_names`: Each row maps the MID of a Freebase object to its textual label.
  - `domains_id_label`: Each row maps the MID of a Freebase domain to its label.
  - `types_id_label`: Each row maps the MID of a Freebase type to its label.
  - `entities_id_label`: Each row maps the MID of a Freebase entity to its label.
  - `properties_id_label`: Each row maps the MID of a Freebase property to its label.
- Subject matter triples file: `fb`x, where $x \in 1, 2, 3, 4$. For each variant, depending on the nature of a task, one can choose to use one of these. For example, to exclude reverse relations but to retain CVT nodes, one can use table `fb3`. All four variants are explained in Section 6 of the paper.
- Type system file: `freebase_endtypes`. This table is built to provide the type system (Section 3 of the paper) for the dataset. Each row in this table maps an edge type to its required subject type and object type.

We also provided three types of scripts for URI simplification (parse_triples.sh), metadata separation (`FBDataDump.sh`), and processing the subject matter triples (`fb`x`.sh`, where $x \in 1, 2, 3, 4$). These scripts are used to generate the aforementioned four variants (discussed in Section 6) and their type systems.

C Basic statistics of the datasets

Statistics of our datasets can be found in Table 1 of the paper.

D Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Described in Section A.3 above.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Described in Section A.4.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments (e.g. for benchmarks)...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Provided at the URL described in Section B.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] They are explained in Section 7 of the paper and at the URL provided in Section B.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] They are explained in Section 7 of the paper and at the URL provided in Section B.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] We have cited all the datasets, frameworks, models, and previous work.

(b) Did you mention the license of the assets? [Yes] It is explained in Section A.1 as well as at the URL in Section B.

(c) Did you include any new assets either in the supplemental material or as a URL? [Yes] The datasets and data preprocessing scripts are made publicly available at the URL in Section B.

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]