FB15k-CVT: A Challenging Dataset for Knowledge Graph Embedding Models

Mouloud Iferroudjene[1,2], Victor Charpenay[1] and Antoine Zimmermann[1]

[1]Mines Saint-Etienne, Univ Clermont Auvergne, INP Clermont Auvergne, CNRS, UMR 6158 LIMOS, Saint-Etienne, France
[2]COURBON Software, 70 rue de la Montat – CS 60327, 42015 Saint-Etienne, France

Abstract

Knowledge Graphs (KGs) are an essential component of neuro-symbolic AI. KG Embedding Models (KGEMs) are used to represent elements of a KG (its entities and relations) in a vector space, to enable efficient processing and reasoning over knowledge. Most KGEMs are evaluated against datasets derived from the Freebase KG: FB15k and FB15k-237. In this paper, we identify limitations in these datasets with respect to Compound Value Types (CVTs), which are nodes introduced in Freebase as a substitute for n-ary relations. In FB15k and FB51k-237, CVTs have been removed, thereby eliminating valuable information. To evaluate whether KGEMs can learn semantically accurate representations of entities and relations in Freebase, we introduce here a new dataset named FB15k-CVT, which reintroduces the deleted CVT nodes. In a preliminary evaluation, we assess the limitations of baseline KGEMs (TransE, DistMult) in the presence of CVTs. The evaluation suggests that KGEMs based on tensor decomposition are more promising than translational models but, most of all, it calls for further experiments with KGEMs that can answer conjunctive queries or that preserve logical entailment.

Keywords

Knowledge Graphs, Neurosymbolic AI, Knowldge Graph Embeddings Models, FB15K-237

1. Introduction

Knowledge graphs (KGs) have become an essential component of neuro-symbolic AI research. A KG is a uniform source of information in which physical-world entities are represented as vertices of a directed edge-labeled graph. In the context of representation learning, edge labels of a KG are called relations, and its edges are called facts or triples [1].

KGs can be leveraged in a great variety of AI applications. Over the past decade, many KG Embedding Models (KGEMs) have been developed for that purpose [1, Sec. 4.2]. By representing entities and relations as numeric structures in a vector space, KGEMs provide a way to integrate both symbolic and sub-symbolic knowledge, enabling efficient processing and reasoning over complex and heterogeneous data. Most KGEMs are evaluated against datasets that are derived from Freebase[1], a (now archived) public KG containing millions of entities and billions of facts.

In KGEM research, the most notable datasets derived from Freebase are FB15k and FB15k-237. FB15k is a subset of Freebase that includes 15k entities selected among the most frequent entities

NeSy 2023, 17th International Workshop on Neural-Symbolic Learning and Reasoning, Certosa di Pontignano, Siena, Italy

mouloud.iferroudjene@emse.fr (M. Iferroudjene); victor.charpenay@emse.fr (V. Charpenay); antoine.zimmermann@emse.fr (A. Zimmermann)

[1]Freebase data dumps https://developers.google.com/freebase, accessed 20 March 2023.

(wrt the number of facts they are associated with) [2]. It was later noticed that FB15k was flawed wrt link prediction (LP), the main learning task for which KGEMs are trained. Many facts used for testing could trivially be reconstructed from other facts exposed to models during training (test leakage) [3]. Among the 1,000+ relations of FB15k, 237 relations were kept in FB15k-237.

Both FB15k and FB15k-237 were designed to be representative subsets of Freebase. Yet, they hide significant details of Freebase that are rarely discussed in KGEM research. Most of the relations in these datasets are actually not in Freebase. 144 relations, out of the 237 of FB15k-237, are not atomic relations but two-hops paths that have been created by composing two Freebase relations. In this paper, we motivate the (re)introduction of atomic relations in FB15k, to be more representative of Freebase—and, in fact, of most large KGs—and describe the resulting dataset.

Our dataset, which we call FB15k-CVT, gets its name from a particular type of entity in Freebase: *Compound Value Types* (CVTs). We introduce CVTs in Sec. 2 and show the limitations that pertain to removing CVTs from Freebase. Fatemi *et al.* already observed that CVTs represented most of Freebase [4]. However, in this paper, we draw different conclusions as to what KGEM research should focus on. While Fatemi *et al.* consider that CVTs should be hidden behind n-ary relations, we argue that the numerous CVTs in Freebase are an ideal benchmark for emerging KGEMs targeting conjunctive query evaluation (such as QUERY2BOX [5]) and logical entailment (such as ELEm [6]). We describe FB15k-CVT in Sec. 3. We give preliminary results with baseline KGEMs for an extension of the link prediction task to two-hops paths in Sec. 4, showing that FB15k-CVT is more challenging than FB15k-237.

2. Compound Value Types in Freebase

Most facts in Freebase are to be interpreted as n-ary statements. Because a typical KG can only capture binary statements, relations with higher arity are represented in Freebase via CVTs [7]. A CVT is an entity whose existence depends upon (at least two) other entities. At a theoretical level, it can be interpreted as the reification of a fact over n entities ($n \geq 2$) and possibly literal values. Examples of CVTs include: a position held by a person over a certain period; a monetary amount; an award nomination relating one or more nominees, an award and a film. The first two examples are given in Fig. 1. Two examples of award nomination CVTs are given in more details in Fig. 1. In Freebase, CVTs may only be defined with respect to non-CVT entities, which are referred to as *topics*. In the examples of Fig 1, 'Dominique de Villepin', 'Minister of Foreign Affairs', 'STM' and 'US dollar' are all topics.

Dealing with n-ary relations has been a long-standing issue on the Semantic Web. Other KGs, including Wikidata [8] and YAGO [9] also include n-ary relations, mostly in the form of spatial-temporal annotations. In fact, the three examples of CVTs we give here also hold temporal annotations. The modeling choice of Freebase, YAGO and Wikidata slightly differ as to how to represent these n-ary annotations but all three KGs use a form of reification that follows documented best practices [10]. CVTs are thus representative of how large KGs represent annotated facts.

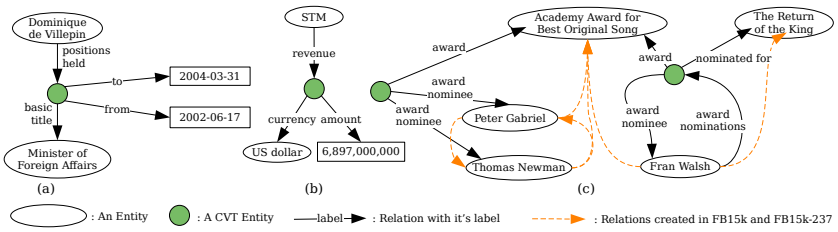

Figure 1: Examples of CVTs in Freebase (circle-shaped green nodes) and the relations created in FB15k and FB15k-237 (dashed orange lines)

2.1. Removing Compound Value Types

The original FB15k dataset was designed for link prediction experiments in a *transductive* setting. That is, a fact $\langle e_h, r, e_t \rangle$ may only be predicted for head and tail entities that are known beforehand. In large open KGs, it is reasonable to assume that the majority of entities has a known identifier (for instance, a Wikipedia page title). It is however less obvious that identifiers for CVTs should be known in advance, given the potentially large number of CVTs in a KG. To train the first KGEMs on Freebase in a transductive setting, Bordes *et al.* chose to collapse CVTs found in Freebase in order to learn embeddings for topics only. As a result, KGEMs are built from a simplification of Freebase that considers only simple relations or paths (of length 2) that bypass CVTs. The simplification process comes with a number of issues.

First, not all paths have been preserved in FB15k. For example, consider the ternary relation `award_nomination` that relates a nominee, a type of award and a creative work. Freebase includes the statement that Fran Walsh was nominated for an Academy Award for Best Original Song in *The Return of the King*, the last film in the Lord of the Rings trilogy (see Fig. 1 on the right). There are three possible paths for this statement: (i) between Fran Walsh and the film, (ii) between Fran Walsh and the award type and (iii) between the film and the award type. FB15k only captures the first two paths. There is no relation between *The Return of the King* and the Academy Award in the dataset. It is unclear what criteria were chosen to decide what paths are preserved and what paths are deliberately lost in the simplification.

To overcome this problem of information loss, Fatemi *et al.* [4] proposed to get rid of CVTs and see FB15k as a hypergraph instead. They created the m-FB15k dataset[2], which includes binary, ternary and quaternary relations. m-FB15k does not include the `award_nomination` relation, though, which is likely due to the incompleteness of Freebase with respect to nominations. For instance, Freebase does not say for what film Peter Gabriel and Thomas Newman were nominated. Turning the CVT into an `award_nomination` statement would generate an empty assignment for `nominated_for`. m-FB15k only includes fully instantiated statements. If KGEMs learn representations of CVTs instead of n-ary statements, they may still be able to correctly predict the missing link pointing to *WALL-E*. A hypergraph representation of CVTs thus remains limited.

Some CVTs also hold information not present in m-FB15k because they relate to literal values and not to topics. Literal values can e.g. be temporal annotations (time instants or intervals).

[2] https://github.com/ElementAI/HypE

For instance, award nominations are associated with a date in Freebase. A dedicated class of KGEMs target temporal KGs, seen as collections of quads. It is possible to generate a temporal KG from CVTs found in FB15k-237. Such a dataset significantly differs from the classical ICEWS benchmark for temporal KGEMs [11]. Quads inherit the limitations of arbitrary n-ary statements in the presence of incomplete knowledge, though. Generic KGEMs that would learn representations of CVTs may also capture temporal information. Temporal relations between CVTs may be materialized in the dataset for KGEMs, in order to turn temporal reasoning tasks into classical link prediction. The nomination of Fran Walsh (in 2003) e.g. preceded that of Peter Gabriel and Thomas Newman (2009). A temporal link prediction task would consist in predicting events e' such that $\langle e, \texttt{precedes}, e' \rangle$ holds, knowing event e occurred.

2.2. Reintroducing Compound Value Types

As already mentioned, the main reason why CVTs were not taken into account in FB15k is the restriction of transductive link prediction to well-known entities (i.e. topics). If the point was valid in 2013, it may not hold anymore, after ten years of KGEM research. Indeed, in recent years, KGEMs have started being evaluated on other tasks beyond link prediction. In particular, several KGEMs target conjunctive query evaluation and logical entailment, such that it is not necessary anymore to maintain an exhaustive list of embeddings for all possible entities.

The first KGEM to focus on conjunctive queries is the Graph Query Embedding (GQE) model [12]. It was then followed by Query2Box [13], BetaE [14] and, more recently, GammaE [15]. All these models can emulate link prediction by answering path queries that bypass CVTs: they would learn against Freebase (without simplification) and only have embeddings for topics and binary relations that are optimized for answering complex queries instead of predicting single facts.

Another class of KGEMs, designed to preserve logical consistency wrt a logic (e.g. a description logic), has recently emerged. They are primarily targeting the $\mathcal{EL}^{++}$ logic, one that is well used in Semantic Web ontologies. These include EL Embeddings (ELEm) [6], BoxEL [16] and the more recent Box^{2}EL [17], all based on the idea of representing classes of entities as convex polytopes in the embedding space. In KGEMs that preserve the semantics of $\mathcal{EL}^{++}$, logical axioms involving compositions of relations can be formulated *a priori* and guide training. As for query evaluation, no embedding for CVTs would then be required.

Despite their increased expressivity, these two classes of models are still evaluated on FB15k and FB15k-237. Given that these datasets are not faithful to the original KG, the performance of KGEMs on these datasets may not necessarily reflect their performance in real-world applications. We thus introduce FB15k-CVT in the next section and compare it to other datasets often used to evaluate KGEMs.

3. The FB15k-CVT dataset

The FB15k-CVT dataset is a KG that we created by expanding FB15k-237 with additional types of entities and relations that were not present in it. In this section, we describe how we extracted the triples from Freebase, filtered the relations, and split the dataset into train/validation/test

subsets. We also discuss the size of the dataset and how it compares to other datasets derived from FB15k.

3.1. Creation of the set of triples

First, to create FB15k-CVT, we downloaded the Freebase large reified knowledge graph that contains millions of entities and relations. We took the latest version available, which was last updated in August 2015.[3]

Entity Filtering. Next, we filtered the *Freebase* KG triples to extract only those involving entities that appear in the FB15k-237 dataset (referred to as FB15k-237's "topics"), whether as subjects or objects. We further restricted the selection to triples where the entities are identified by *Machine IDentifier* (MID) consisting of /m/ followed by a base-32 unique identifier, such as /m/026t6. This resulted in triples from *Freebase* KG where the "topics" of FB15k-237 are connected to new entities labeled as "CVT". Consequently, the current set of triples contains additional entities and not yet filtred set of relations, which includes many new triples. Among these are the one-hop relations already present in the FB15k-237 KG, such as (/m/01lxd4, /music/genre/artists, /m/0f0y8).

Relation Filtering. Next, to determine which triples to retain, we compile a set of relation filters that includes both the decomposition of two-hop relations and the one-hop relations present in FB15k-237. This set contains a total of 302 relation types. Note that FB15k-237 consists of 237 relations, of which 93 are direct relations and the remaining 144 are composed relations. After, we apply this filter, we obtain a smaller dataset size, with only 14,468 topics, which represents represents 99% of FB15k-237's entities, and 292 types of relations, which corresponds to 96% of the total types of relations found in FB15k-237.

Introduction of CVTs. By filtering the extracted set of triples this way, we introduce "CVT" entities back, where it constitutes the intermediate nodes between *topics*. CVTs were mainly hidden within the composed relations in FB15k-237, e.g., the entities and relations in following two triples : (i) (/m/04ljl_l, /award/award_category/nominees, /CVT/1234); (ii) (/CVT/1234, $/award/award\_nomination/nominated\_for$,/m/03qcfvw) are originated from the following triple with a composed two-hop relation (/m/04ljl_l, /award/award_category/nominees./award/award_nomination/nominated_for, /m/03qcfvw) in FB15k-237 KG.

3.2. Train/validation/test split

Next, we want to ensure that the train/validation/test split of FB15k-CVT gets the same distribution of topics and relations as in FB15k-237. This will help us lay the groundwork for training, evaluating KGEMs and carrying experiments on the new dataset. First, for each subset, we keep triples with direct relations. Then, for triples with composed relations, we look for intermediate nodes of types *CVT*, in the previously built set of triples, to split the relation into two new triples that formulate a path. Each path consists of two triples that include 3 entities and 2 relations. Only one of the entities is of type *CVT*. We note them as follow: e_1, e_2 for the topics, n_{cvt} for the intermediate node and r_1, r_2 for the relations. Depending on the direction of the relations,

[3] http://commondatastorage.googleapis.com/freebase-public/rdf/freebase-rdf-latest.gz

Dataset	#Statements	$\lvert\mathcal{E}_{\text{topic}}\rvert$	$\lvert\mathcal{E}_{\text{CVT}}\rvert$	$\lvert\mathcal{R}\rvert$	#Train	#Valid	#Test
FB15k	592,213	14,951	/	1,345	483,142	50,000	59,071
Fb15k-237	310,079	14,505	/	237	272,115	17,526	20,438
m-FB15k	493,520*	10,314	/	71	415,375*	39,348*	38,797*
YAGO15k	138,056	15,403	/	34	110,441	13,815	13,800
FB15k-CVT	1,501,110	14,468	454,070	292	1,421,877	34,479	44,754

Table 1
Datasets characteristics after preprocessing. (*) statements in m-FB15k are represented with n-ary relations

we distinguished 3 types of paths in our dataset: (i) **direct path**: $(e_1, r_1, n_{\text{cvt}})/(n_{\text{cvt}}, r_2, e_2)$; (ii) **splitting path**: $(e_1, r_1, n_{\text{cvt}})/(e_2, r_2, n_{\text{cvt}})$; (iii) **joining path**: $(n_{\text{cvt}}, r_1, e_1)/(n_{\text{cvt}}, r_2, e_2)$. By following this filtering and splitting process, we ensure that the built dataset remains closely aligned with the original FB15k-237 dataset. The resulting dataset has the characteristics given in Table 1, compared with other reference datasets. It is available online[4].

4. Experiments and preliminary results

In this section, we present the details of the conducted experiments and the preliminary results. We run our evaluation of KGEMs on two datasets: FB15k-237 and our built FB15k-CVT datatset. We focus on two widely used baseline models, TransE and DistMult, and analyze their performance in the presence of CVTs in KGs. The later is assessed on the path prediction (PP) tasks.

4.1. From link prediction to path prediction

Generally, researchers assess the performance of their developed KGMEs on a link prediction (LP) task, which consists of predicting the presence or absence of a relation between two entities in the knowledge graph. In practice, they first train their models using a subset of triples $\mathcal{T}_{train}$ from the knowledge graph $\mathcal{KG}$ dataset i.e., $\mathcal{T}_{train} \subset \mathcal{E} \times \mathcal{R} \times \mathcal{E} \subset \mathcal{KG}$. The goal of this step is to learn vector representations for the set of *entities* $\mathcal{E}$ and *relations* $\mathcal{R}$. After training, a set of unseen evaluation triples is provided $\mathcal{T}_{eval}$, with entities and relations known beforehand. For each triple $(h, r, t) \in \mathcal{T}_{eval}$, they evaluate their models on two different tasks: (i) **head prediction** $(?, r, t)$ and (ii) **tail prediction** $(h, r, ?)$. Technically, a relation r and a tail t (Resp. head h) entity are provided, and the model aims to rank correctly the head h (resp. tail h) *ground-truth* entity, such that $(h, r, t) \in \mathcal{T}_{eval}$. This, after scoring each possible $(h', r, t), \forall h' \in \mathcal{E}$ (resp. $(h, r, t'), \forall t' \in \mathcal{E}$) with the same scoring method as in the training phase.

However, in our case, LP task is not proper for evaluating KGEMs on capturing the complex relations in FB15k-CVT, which involve two-hops paths along with intermediate CVT entities to represent n-ary relations. Therefore, we propose to evaluate KGEMs on a path prediction task, where the goal is to predict a complete path between two topic entities in a knowledge graph. Differently from LP task, a set of unseen evaluation quads, representing paths, is created $\mathcal{Q}_{eval} \subset$

[4] https://seafile.emse.fr/d/d5bad2a20d3b4971be2b/

$\mathcal{E}_{topics} \times \mathcal{R} \times \mathcal{R} \times \mathcal{E}_{topics}$. For each quadruple (e_1, r_1, r_2, e_2), we evaluate the models on four possible PP tasks: (i) **chain backward prediction** i.e., $(?, r_1, n_{\text{cvt}})/(n_{\text{cvt}}, r_2, e_2)$, (ii) **chain forward prediction** i.e., $(e_1, r_1, n_{\text{cvt}})/(n_{\text{cvt}}, r_2, ?)$, (iii) **join prediction** i.e., $(e_1, r_1, n_{\text{cvt}})/(?, r_2, n_{\text{cvt}})$ and (iv) **split prediction** i.e., $(n_{\text{cvt}}, r_1, e_1)/(n_{\text{cvt}}, r_2, ?)$. In other terms, we are given an entity of type topic e_1 (resp. e_2) and two relations r_1 and r_2, and then evaluated on ranking correctly the ground-truth e_2 (resp. e_1). This evaluation task aims at assessing the ability of KGEMs to learn the embedding space of CVTs depending on the relations they are involved in. This provides a more comprehensive and challenging benchmark for evaluating the effectiveness of KGEMs on real-world KGs.

4.2. Experiments Settings

Datasets. In the following experiments, we consider, for now, only the two datasets: FB15k-237 (for reproduction) and our newly built dataset FB15k-CVT (for evaluation). For FB15k-237, we closely reproduced the results in the original papers of the models (TransE [2] and DistMult [18]), assessed on LP tasks. On the other hand, for FB15k-CVT, we evaluated the same KGEMs on path prediction (PP) tasks.

Preprocessing of FB15k-CVT. Now that the split of FB15k-CVT dataset is done, we need to look up and prevent any flaws in it. First, we adapt the dataset for transductive settings; by deleting from validation and test sets the triples with topics, that don't appear not appearing in the training set. This resulted in deleting 4 triples from validation set and 3 others from test set. Secondly, we follow the same stratregy as in [3] to prevent inverse relation leakage . The latter was between triples that contain at least on entity of type CVT, i.e., $\{(?e_1, ?r_1, ?e_2)_{\text{train}} \cap (?e_2, ?r_2, ?e_1)_{\text{eval}}\}$, where $r_1 \neq r_2$. Finally, we wrap up by deleting any duplicate triples remaining between evaluation sets (i.e., validation and test sets) and training set. The goal of this preprocessing step is to create a more challenging dataset that requires models to generalize better to new relation between entities.

Creating paths (quadruples) for validation and test sets As stated in subsection.4.1, we recall that our PP task requires a set of quadruples, noted $\mathcal{Q}_{\text{Eval}}$,in order evaluate of the models. Therefore, we generate all possible path in our subsets and delete the intermediate CVT nodes. As a result, we got $|\mathcal{Q}_{\text{test}}| = 8{,}667$ quadruples. Also, it's worth noting that along this process, we make certain there is no paths with a reflexive relation, i.e., $(e, r, n_{\text{cvt}}), (n_{\text{cvt}}, r, e)$ as this will bias the results of the evaluation.

Model training. We choose two KGE models, TransE [2] and DistMult [18], which belongs respectively to different sub-categories of tensor-based models: translational and tensor decomposition models, to train on $\mathcal{T}_{\text{train}}$ of the chosen datasets. For this, we used the best configurations we found, as provided in the PyKEEN framework[5], designed for FB15k-237 and reused them for FB15k-CVT.

Model Evaluation. In our evaluation setting for PP task, we restrict the models to only to use the learned embeddings for topic $\mathcal{E}_{topic}$ and $\mathcal{R}$. As we consider that not all CVTs are seen by the model in training set. To this end, in this phase, we slightly modified the scoring functions of the evaluated models, i.e., TransE and DistMult, to suit the evaluation task (see eq. 1 and

[5]PyKEEN (Python KnowlEdge EmbeddiNgs) https://pykeen.github.io/

Model		TransE				DistMult			
Dataset	Task	MR	Hit@1	Hit@5	Hit@10	MR	Hit@1	Hit@5	Hit@10
FB15k-237	LP (Both)	192.52	0.1640	0.3641	0.4626	777.79	0.1811	0.3102	0.3708
FB15k-CVT	PP (All)	237.18	0.0709	0.0994	0.1142	139.81	0.0933	0.1442	0.1646

Table 2
Evaluation's results of TransE and DistMult model on FB15k-237 (Reproduction) and FB15k-CVT datasets.*All*

eq. 2). Note that the function $\text{sign}(r)$ returns the direction of the relation r, in order to calculate accurately the projection of the pair (e, r) to the region, in the latent space, representing the CVTs embeddings.

$$f_{TransE}(e_1, r_1, r_2, e_2) = \|(\vec{e_1} + \text{sign}(r_1).\vec{r_1}) - (\vec{e_2} - \text{sign}(r_2).\vec{r_2})\| \quad (1)$$

$$f_{DistMult}(e_1, r_1, r_2, e_2) = \|(\vec{e_1} \odot \text{sign}(r_1).\vec{r_1}^D) - (\vec{e_2} \odot \text{sign}(r_2).\vec{r_2}^D)\| \quad (2)$$

Evaluation Metrics. As we evaluated the models on link-like prediction tasks. There are two type of metric in the literature: ranking and classification metrics, to quantify the models' performance. For our experiments, among various available metrics we focus on rank-based ones to measure different aspects of ranking performance. Precisely, we choose: (1) *Mean Rank* (MR) metric, which is a base metric that summarizes the central tendency of ranks by computing the arithmetic mean over all individual ranks, and (2) *The Hits at K* metric, noted Hit@k, which measures the fraction of times when the correct result is in the top-k ranked triples. We set k for our experiments to be $k \in \{1, 5, 10\}$.

4.3. Experiments Results

Table 2 summarizes the performance of TransE and DistMult models on FB15k-237 and FB15k-CVT, respectively, for the link prediction (LP) and path prediction (PP) tasks. We report the average results for the models' performance on sub-prediction tasks, including head and tail prediction for LP, as well as the four subtasks mentioned in subsection 4.1 for PP. In the first line, both models exhibit, relatively to stat-of-the-art link prediction KGEMS, high Hits@k. In second line, for istance, one could observe that TransE gets high mean rank (MR) and low Hit@k. This indicates that the model is not well suited for path prediction in the presence of CVTs. Detailed results show that it is especially the case for the "chain forward" and "chain backward" predictions (see Table 3 in Appendix for more details). Results for DistMult are slightly more contrasted. We observe the same pattern but the decrease in performance is much lower. MR even significantly improves for DistMult.

However, the evaluation confirms that both models got challenged in path prediction task, as they couldn't succeed at learning the underlying vector representation of CVTs. Furthermore, the evaluation calls for further experiements with KGEMs with our proposed FB15k-CVT dataset to explore new avenues of existing models or improving or developing new ones.

5. Conclusion

The FB15k-CVT dataset that we have described was generated from FB15k and its successor, FB15k-237. Its main distinctive characteristic is that it is an exact subset of Freebase, which FB15k and FB15k-237 aren't. We argued in the paper that the reintroduction of CVTs in the dataset offers new challenges for emerging KGEMs, including Query2Box, ELEm and their competitors. Our preliminary evaluation of TransE and DistMult shows that the overall performances of the two models significantly decrease in the presence of CVTs but the change benefits DistMult, suggesting that tensor decomposition is better suited for handling CVTs than translational models.

References

[1] A. Hogan, E. Blomqvist, M. Cochez, C. D'amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A.-C. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, A. Zimmermann, Knowledge graphs, ACM Comput. Surv. 54 (2021). URL: https://doi.org/10.1145/3447772. doi:10.1145/3447772.

[2] A. Bordes, N. Usunier, A. García-Durán, J. Weston, O. Yakhnenko, Translating Embeddings for Modeling Multi-relational Data, in: C. J. C. Burges, L. Bottou, Z. Ghahramani, K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States, 2013, pp. 2787–2795. URL: https://proceedings.neurips.cc/paper/2013/hash/1cecc7a77928ca8133fa24680a88d2f9-Abstract.html.

[3] K. Toutanova, D. 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, Association for Computational Linguistics, Beijing, China, 2015, pp. 57–66. URL: https://aclanthology.org/W15-4007. doi:10.18653/v1/W15-4007.

[4] B. Fatemi, P. Taslakian, D. Vazquez, D. Poole, Knowledge Hypergraphs: Prediction Beyond Binary Relations, 2020. URL: http://arxiv.org/abs/1906.00137, arXiv:1906.00137 [cs, stat].

[5] H. Ren, W. Hu, J. Leskovec, Query2box: Reasoning over Knowledge Graphs in Vector Space using Box Embeddings, 2020. URL: http://arxiv.org/abs/2002.05969, arXiv:2002.05969 [cs, stat].

[6] M. Kulmanov, W. Liu-Wei, Y. Yan, R. Hoehndorf, EL Embeddings: Geometric Construction of Models for the Description Logic EL++, in: S. Kraus (Ed.), Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019, ijcai.org, 2019, pp. 6103–6109. URL: https://doi.org/10.24963/ijcai.2019/845.

[7] T. Pellissier Tanon, D. Vrandečić, S. Schaffert, T. Steiner, L. Pintscher, From freebase to wikidata: The great migration, in: Proceedings of the 25th International Conference on World Wide Web, WWW '16, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 2016, p. 1419–1428. URL: https://doi.org/10.1145/2872427.2874809. doi:10.1145/2872427.2874809.

[8] D. Vrandečić, M. Krötzsch, Wikidata: A free collaborative knowledgebase, Commun. ACM 57 (2014) 78–85. URL: https://doi.org/10.1145/2629489. doi:10.1145/2629489.
[9] J. Hoffart, F. M. Suchanek, K. Berberich, G. Weikum, Yago2: A spatially and temporally enhanced knowledge base from wikipedia, Artificial Intelligence 194 (2013) 28–61. URL: https://www.sciencedirect.com/science/article/pii/S0004370212000719. doi:https://doi.org/10.1016/j.artint.2012.06.001, artificial Intelligence, Wikipedia and Semi-Structured Resources.
[10] N. Fridman Noy, A. L. Rector, Defining N-ary Relations on the Semantic Web, W3C Note, World Wide Web Consortium, 2006. URL: https://www.w3.org/TR/2006/NOTE-swbp-n-aryRelations-20060412/.
[11] A. García-Durán, S. Dumančić, M. Niepert, Learning Sequence Encoders for Temporal Knowledge Graph Completion, 2018. URL: http://arxiv.org/abs/1809.03202, arXiv:1809.03202 [cs].
[12] W. L. Hamilton, P. Bajaj, M. Zitnik, D. Jurafsky, J. Leskovec, Embedding Logical Queries on Knowledge Graphs, in: S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett (Eds.), Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada, 2018, pp. 2030–2041. URL: https://proceedings.neurips.cc/paper/2018/hash/ef50c335cca9f340bde656363ebd02fd-Abstract.html.
[13] H. Ren, W. Hu, J. Leskovec, Query2box: Reasoning over Knowledge Graphs in Vector Space using Box Embeddings, 2020. ArXiv:2002.05969 [cs, stat].
[14] H. Ren, J. Leskovec, Beta Embeddings for Multi-Hop Logical Reasoning in Knowledge Graphs, in: Advances in Neural Information Processing Systems, volume 33, Curran Associates, Inc., 2020, pp. 19716–19726. URL: https://proceedings.neurips.cc/paper/2020/hash/e43739bba7cdb577e9e3e4e42447f5a5-Abstract.html.
[15] D. Yang, P. Qing, Y. Li, H. Lu, X. Lin, GammaE: Gamma Embeddings for Logical Queries on Knowledge Graphs, in: Y. Goldberg, Z. Kozareva, Y. Zhang (Eds.), Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022, Association for Computational Linguistics, 2022, pp. 745–760. URL: https://aclanthology.org/2022.emnlp-main.47.
[16] B. Xiong, N. Potyka, T. Tran, M. Nayyeri, S. Staab, Faithful Embeddings for $\mathcal{EL}^{++}$ Knowledge Bases, in: U. Sattler, A. Hogan, C. M. Keet, V. Presutti, J. P. A. Almeida, H. Takeda, P. Monnin, G. Pirrò, C. d'Amato (Eds.), The Semantic Web - ISWC 2022 - 21st International Semantic Web Conference, Virtual Event, October 23-27, 2022, Proceedings, volume 13489 of *Lecture Notes in Computer Science*, Springer, 2022, pp. 22–38. URL: https://doi.org/10.1007/978-3-031-19433-7_2.
[17] M. Jackermeier, J. Chen, I. Horrocks, Box^{2}EL: Concept and Role Box Embeddings for the Description Logic EL++, 2023. URL: http://arxiv.org/abs/2301.11118, arXiv:2301.11118 [cs].
[18] B. Yang, W.-t. Yih, X. He, J. Gao, L. Deng, Embedding Entities and Relations for Learning and Inference in Knowledge Bases, in: Y. Bengio, Y. LeCun (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015. URL: http://arxiv.org/abs/1412.6575.
[19] R. Abboud, Learning and inference over relational data, PhD Thesis, Uni-

versity of Oxford, Oxford, 2022. URL: https://ora.ox.ac.uk/objects/uuid:da7744ad-effd-4fc9-b7ab-a00b03a86a53.

[20] P. Wang, J. Chen, L. Su, Z. Wang, N-ary relation prediction based on knowledge graphs with important entity detection, Expert Systems with Applications 221 (2023) 119755. URL: https://linkinghub.elsevier.com/retrieve/pii/S0957417423002567. doi:10.1016/j.eswa.2023.119755.

[21] J. Leblay, M. W. Chekol, Deriving Validity Time in Knowledge Graph, in: Companion of the The Web Conference 2018 on The Web Conference 2018 - WWW '18, ACM Press, Lyon, France, 2018, pp. 1771–1776. URL: http://dl.acm.org/citation.cfm?doid=3184558.3191639. doi:10.1145/3184558.3191639.

[22] T. Lacroix, G. Obozinski, N. Usunier, Tensor Decompositions for temporal knowledge base completion, 2020. URL: http://arxiv.org/abs/2004.04926, arXiv:2004.04926 [cs, stat].

[23] J. Messner, R. Abboud, I. I. Ceylan, Temporal Knowledge Graph Completion Using Box Embeddings, Proceedings of the AAAI Conference on Artificial Intelligence 36 (2022) 7779–7787. URL: https://ojs.aaai.org/index.php/AAAI/article/view/20746. doi:10.1609/aaai.v36i7.20746.

[24] M. Ali, M. Berrendorf, C. T. Hoyt, L. Vermue, M. Galkin, S. Sharifzadeh, A. Fischer, V. Tresp, J. Lehmann, Bringing Light Into the Dark: A Large-scale Evaluation of Knowledge Graph Embedding Models under a Unified Framework, IEEE Transactions on Pattern Analysis and Machine Intelligence (2021) 1–1. URL: https://ieeexplore.ieee.org/document/9601281/. doi:10.1109/TPAMI.2021.3124805.

[25] Z. Sun, Z.-H. Deng, J.-Y. Nie, J. Tang, RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space, 2019. URL: http://arxiv.org/abs/1902.10197, arXiv:1902.10197 [cs, stat].

[26] S. Guo, Q. Wang, L. Wang, B. Wang, L. Guo, Jointly Embedding Knowledge Graphs and Logical Rules, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Austin, Texas, 2016, pp. 192–202. URL: http://aclweb.org/anthology/D16-1019. doi:10.18653/v1/D16-1019.

[27] W. Zhang, B. Paudel, L. Wang, J. Chen, H. Zhu, W. Zhang, A. Bernstein, H. Chen, Iteratively Learning Embeddings and Rules for Knowledge Graph Reasoning, in: The World Wide Web Conference, ACM, San Francisco CA USA, 2019, pp. 2366–2377. URL: https://dl.acm.org/doi/10.1145/3308558.3313612. doi:10.1145/3308558.3313612.

[28] J. Lajus, L. Galárraga, F. Suchanek, Fast and Exact Rule Mining with AMIE 3, in: A. Harth, S. Kirrane, A.-C. Ngonga Ngomo, H. Paulheim, A. Rula, A. L. Gentile, P. Haase, M. Cochez (Eds.), The Semantic Web, volume 12123, Springer International Publishing, Cham, 2020, pp. 36–52. URL: http://link.springer.com/10.1007/978-3-030-49461-2_3. doi:10.1007/978-3-030-49461-2_3.

[29] C. Meilicke, M. W. Chekol, D. Ruffinelli, H. Stuckenschmidt, Anytime Bottom-Up Rule Learning for Knowledge Graph Completion, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, International Joint Conferences on Artificial Intelligence Organization, Macao, China, 2019, pp. 3137–3143. URL: https://www.ijcai.org/proceedings/2019/435. doi:10.24963/ijcai.2019/435.

[30] S. Ott, C. Meilicke, M. Samwald, SAFRAN: An interpretable, rule-based link prediction method outperforming embedding models, 2021. URL: http://arxiv.org/abs/2109.08002,

arXiv:2109.08002 [cs].

TransE

Dataset	Task	MR	Hit@1	Hit@3	Hit@5	Hit@10
FB15k-237	Head	257.29	0.1025	0.2123	0.2724	0.3654
	Tail	127.75	0.2256	0.3800	0.4558	0.5597
	Both	192.52	0.1640	0.2961	0.3641	0.4626
FB15k-CVT	Chain forward	235.80	0.0472	0.0635	0.0714	0.0826
	Chain Backward	248.86	0.0396	0.0570	0.0690	0.0857
	Join	148.20	0.4302	0.4613	0.4823	0.5082
	Split	196.12	0.3902	0.4188	0.4381	0.4619
	All	237.18	0.0709	0.0887	0.0994	0.1142

DistMult

Dataset	Task	MR	Hit@1	Hit@3	Hit@5	Hit@10
FB15k-237	Head	1047.92	0.0858	0.1469	0.1820	0.2348
	Tail	507.66	0.2763	0.3867	0.4383	0.5068
	Both	777.79	0.1811	0.2668	0.3102	0.3708
FB15k-CVT	Chain forward	148.04	0.0447	0.0611	0.0696	0.0852
	Chain Backward	143.19	0.0863	0.1391	0.1554	0.1775
	Join	74.21	0.4573	0.5085	0.5376	0.5664
	Split	63.65	0.4154	0.4711	0.5296	0.5793
	All	139.81	0.0933	0.1293	0.1442	0.1646

Table 3
Evaluation's results of TransE and DistMult model on FB15k-237 (Reproduction) and FB15k-CVT datasets

A. Extended results of the experiment

We provide detailed results in Table 3.

B. Related Work

As argued in Sec. 2, our dataset brings new challenges to KGEMs that intend to capture either n-ary statements or rules. We now review existing approaches for these two cases and the datasets on which they were evaluated.

HypE was the first model to target n-ary statements via hypergraph embeddings [4]. It was then followed by several models, including extensions of TransE [2] and DistMult [18], BoxE [19, p. 67] and more recent approaches based on graph neural networks (such as Att-ImpGCN [20]). These models are evaluated against m-FB15k, which is derived from FB15k and not FB15k-237. The separation between training, validation and testing in FB15k is known to lead to data leakage, hence the creation of FB15k-237 [3]. The problem, which is due to inverse relations not being taken into account, persists in the n-ary variant. The results of some models, including SimplE, are artificially high. HypE is derived from SimplE and should be reevaluated against our dataset. The same applies to similar methods based on tensor decomposition.

As already mentioned, temporal relations are a special case of n-ary relations. Temporally aware TransE and DistMult (referred to as TA-TranE and TA-DistMult), are among the early

KGEMs designed for temporal relations [11]. They were tested on YAGO15k[6], a temporal KG derived from FB15k, by aligning entities with YAGO and retrieving YAGO facts that included 'occurs since' and 'occurs until' annotations. TA-TransE and TA-DistMult were also evaluated on a much larger dataset derived from Wikidata. This dataset was originally constructed to evaluate TTransE, a temporal extension of TransE [21]. Temporal annotations in the Wikidata dataset (and most of the YAGO15k dataset) are only at coarse granularity: statements are annotated with years. A larger temporal KG, also derived from Wikidata, was used to evaluate temporal extensions of ComplEx (TComplEx and TNTComplEx) [22]. The authors of these extensions considerd YAGO15k was too small for temporal link prediction. More recent temporally aware KGEMs, such as BoxTE [23], tend to favor other benchmarks over YAGO15k.

At another end of the spectrum, some KGEMs focus on rule induction. For instance, RotatE, one of the best performing KGEMs on FB15k-237[7], was in fact motivated by the ability to capture composition rules, as well as symmetry, antisymmetry and inversion [25]. Before that, the Knowledge and Logic Embedding method (KALE), integrated rules to TransE using fuzzy logic expressions [26]. For evaluation, 47 rules were defined for a subset of FB15k. These rules have a restricted form, though: they are limited to subsumption and (basic) composition. KALE assumes rules are provided. In contrast, the IterE model is designed to learn rules jointly with entity and relation representations [27]. Rules correspond to linear algebraic expressions in the embedding space, such that the plausibility of a rule can be calculated as a distance between matrices. IterE can learn the same rules as those targeted by RotatE but it can e.g. not learn subsumption.

KALE, IterE and other rule learning models generally focus on restricted rule forms, so that the space of possible inductions is of reasonable size. In fact, if one specifically looks for Horn rules, exploration may be performed directly in the syntactic space: rules may be mined without leverage any latent embedding space. It is the approach taken e.g. by the latest Association Rule Mining under Incomplete Evidence (AMIE 3) [28] and Anytime Bottom-up Rule Learning (AnyBURL) [29] algorithms. On link prediction, AnyBURL tends to have comparable results with standard KGEMs over FB15k-237 [29, 30]. It may in fact outperform them on our dataset, given the high support rules involving CVT values have.

[6] training, validation, and test splits are available at available at https://github.com/mniepert/mmkb/tree/master/TemporalKGs/yago15k
[7] as reported in a large-scale experiment by Ali *et al.* [24].