Towards Understanding the Impact of Graph Structure on Knowledge Graph Embeddings

Anonymous authors.

No Institute Given

Abstract. Knowledge graphs are an established paradigm for integrating heterogeneous data and representing knowledge. As such, there are many different methodologies for producing knowledge graphs, which span notions of expressivity, and are tailored for different use-cases and domains, as well as different technological frameworks (e.g., labeled property graphs). Now, as neurosymbolic methods rise in prominence, it is important to understand how the development of knowledge graphs according to these methodologies impact downstream tasks, such as knowledge graph embeddings. In this paper, we conduct several experiments on modified versions of the FB15k-237 knowledge graph to understand these impacts and discuss the performance of different KGE models.

1 Introduction

Knowledge graphs (KGs) are, by now, an established paradigm for integrating heterogeneous data [20] with quite a lot of history [11]. Over the years, many different methodologies for the development of a KG have surfaced. The earliest are drawn from early ontology literature [9,35] and see an evolution in complexity as best practices surface, especially in the face of maturing use-cases [27]. Most recently, from the ontology side, we see pattern-based methods [6,25] to create semantically-rich, yet easily composed ontologies to act as a *schema* for a KG. On the other hand, we see a burgeoning interest in the use of upper ontology (namely BFO [26]) as an emerging standard in the US military-industrial complex [1], as well as more broadly defined [17]. Finally, in the private sector, we see an emerging bloc dedicated to the use of labeled property graphs (LPGs) [4,33]. While these types of graphs may be arbitrarily interoperable with "classic" RDF KGs, until RDF-star [22] reaches W3C recommendation status, there is currently no agreed upon way to enforce this.

These different methodologies produce KGs that can all solve the same problems, but do so in different ways, and thus meet different needs of different stakeholders. While this is not particularly concerning in and of itself, it is important to assess how they perform with respect to downstream tasks, such as creating a knowledge graph emebedding (KGE) [16,14]. These KGEs can be used in (non-logical) inference tasks, for computing, e.g., prediction of links or relations between entities [23], similarity between entities [29], and clustering of entities [24].

Historically speaking, the standard metric for measuring quality of an ontology cum KG would be to execute SPARQL queries, which correspond to competency questions (CQs) [5], against the KG and assess if the result is sufficient for the use-case. However,

with the advent (and maturation) of the use of KGEs, it behooves us to expand our notion of quality, and understand the mechanisms behind performant KGEs as they pertain to high quality KGs (i.e., those with schemas).

Some preliminary work revealed that altering the conceptualization of a schema (i.e., rich vs. shallow semantics) leads to differences in the performance of KGEs [8]. While the change in performance was marginal (on the order of 10^{-2}), it was consistent across most KGE models and datasets. This has lead us to the current work: we wish to examine how following best practices for KG development, understood as the rigorous development of an ontology to be used as a schema, impacts the ability to train a KGE model, which KGE models perform differently (and why) on different graph structures, and which KGE tasks have the best performance across the spectrum of graph structures. To the authors' knowledge, we do not know of any systematic attempts to understand this.

Essentially, we wish to know if the incorporation of a schema into a KG is actually helpful and to what extent. If it is not, why not, and how can existing KGE models be adapted or new ones be developed. This paper presents our work to begin answering these questions. Specifically, we have taken the classic FB15k-237 dataset [31] and provided a series of augmentations that progressively mimic the inclusion of a schema into the KG. These new datasets we call FB15k-238 and -239. We assess the performance using KGE models as implemented in the DGL-KE [36] library. Concretely,

1. the augmented FB15k-237 datasets: FB15k-238 and FB15k-239,
2. the scripts and configuration to generate these datasets,
3. a thorough evaluation of the effects that the incorporation of increasing metadata has on the performance of the KGE models; and
4. a brief discussion of the possible underlying effects.

The rest of this paper is organized as follows. Section 2 provides a foundational basis for the paper, our augmented FB15k-237 dataset, and related work. In Sections 3, 4, and 5, we present our experiment methodology, our results, and a discussion thereof. Finally, in Section 6, we conclude and present our next steps.

2 Background

In this section, we briefly discuss the different KGE models used in our experiments, note our use of the FB15k-237 dataset, and state the related work we were able to find regarding this effort.

2.1 Knowledge Graph Embedding Methods

This section discusses the techniques for KGE which involve associating entities (head and tail, respectively denoted as *h* and *t*) and relationships (denoted as *r*) as vectors within a mathematical hyperplane combined with machine learning techniques to create the respective models, which the experiment utilizes from the DGL-KE library. Table 1 depicts models alongside their respective scoring functions.

Model	Scoring Function $f(h, r, t)$
TransE	$-\|\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|\|_p$
TransR	$\|\|\mathbf{h_r} + \mathbf{r} - \mathbf{t_r}\|\|_2^2$
RotatE	$-\|\|h \odot r - t\|\|$
RESCAL	$\mathbf{h}^\top \mathbf{W}_r \mathbf{t} = \sum_{i=1}^{d} \sum_{j=1}^{d} w_{ij}^r h_i t_j$
DistMult	$\mathbf{h}^\top \mathbf{W}_r \mathbf{t} = \sum_{i=1}^{d} \mathbf{h}_i \cdot diag(\mathbf{W}_r)_i \cdot \mathbf{t}_i$
ComplEx	$Real(h \odot r \odot t)$

Table 1: The six KGE Models we use – at a glance. $\odot$ is used to denote the Hadamard product.

2.1.1 Translational Distance Models Translational Distance Models depict entities and relationships within a vector space, also referred to as an embedding space. These models' scoring functions are functions of the distance in the embedding spaces.

TransE [7] specializes in the translation of head to tail embeddings as a simple approach for relational extraction. TransE allows for understanding hierarchical relationships as the embedding space would illustrate that a sibling nodes should be in close proximity of one another on *one axis* and the parent-child relationship would be depicted as a translation on *another axis*. While the TransE methodology is straightforward, an observed drawback is the model's application only being suitable for 1-to-1 relationships. The scoring function is described as the distance between entity *h* and relationship *r* to entity *t*.

TransR [18] is designed to handle multiplicity in relationships by separating the relationships from entities in the vector space; thus, allowing for a representation of the relationships within their own embedding space. By having an embedding space that describes entities and relationships independently, the entity embedding space can be projected onto the relation embedding space and relationship *r* can be calculated as the distance from head and tail. The scoring function is similar to TransE; however, relates entity *h* and *t* with a corresponding relationship *r*.

RotatE [28] is also designed similarly to TransE by representing entities within a vector space; however, with the addition of complex values on the entities to explore relations as rotations within the complex entity vector space. Motivated by Euler's formula, which represents complex numbers in a circular unit, RotatE aims to formally model and infer a knowledge graph's relational symmetry, inverse, and composition.

2.1.2 Semantic Matching Models Semantic Matching Models utilize a similarity-based scoring function to capture latent semantics of a knowledge graph.

RESCAL [19] is a bilinear model that depicts entities and their relationships as a three-way tensors. RESCAL is a multi-step process designed to dyadically learn and understand the interconnected relationship between entities by performing *collective learning*. By breaking down the entity and relationship tensors into smaller components, RESCAL is able to learn latent semantic within the knowledge graph.

DistMult [34] simplifies RESCAL by using a diagonal matrix restricting the dimensions of relationships to be equal to the parameters equal to TransE by preventing values outside of the diagonal of a matrix from containing non-zero values. The limitations of DistMult are observed as not having the ability to handle antisymmetric relationships between a head and tail entity.

ComplEx [32] extends DistMult by using complex values to allow for a representation of symmetry and antisymmetry in the representation of entities and relationship matrices. By extracting the real values from the imaginary values from the respective matrix, ComplEx is able to represent vector rotations in the vector space similar to RotatE, as the real values would represent symmetric relationships and imaginary values would represent asymmetric relationships.

2.2 FB15k-237

FB15k-237 was introduced in [31] as a subset of Freebase for benchmarking link predictions, specifically analyzing observed and latent features in text inferencing. Table 2 shows the counts of entities and relationships provided FB15k-237. FB15k-237 was designed to challenge KGE models as the dataset does not include inverse relationships, which could leak into a model's link prediction task. The relationships chosen in FB15k-237 was selected from a subset consisting of the most frequent 401 relationships. The 401 relationships were then pruned down to 237 by removing near-duplicate relationships and inverse relations between head and tail entities.

2.3 Related Work

The limitations of the FB15k and FB15k-237 datasets are argued in [15] as the removal of Freebase *Compound Value Types (CVTs)* consequently removes valuable information. *FB15k-CVT* is an exact subset of Freebase with the incorporation of CVTs. CVTs allow knowledge bases to create more structured and detailed representation of entities with multiple values of a type of data. When evaluating KGE models against FB15k-237 and FB15k-CVT, FB15k-CVT underperformed on link prediction tasks. This work indicates that current KGE models may not effectively incorporate semantic data and additional research can be done to understand the limitations.

Overall, we see that deductive reasoning is quite difficult outside of the symbolic algorithms dedicated to it. In particular, neurosymbolic methods (e.g., as found in [13]) struggle quite a bit. As deductive reasoning is a major hurdle for approaching human-level cognition, this provides further motivation for understanding the impact of how the presence (or lack thereof) impacts the formation of latent or embedding spaces.

3 Methodology

In this section, we outline the creation of the new datasets FB15k-238 and FB15k-239 (Sections 3.1 and 3.2, respectively). Table 2 shows the statistics (i.e., number of entities, relationships, and triples overall – and how they were divided among training, testing,

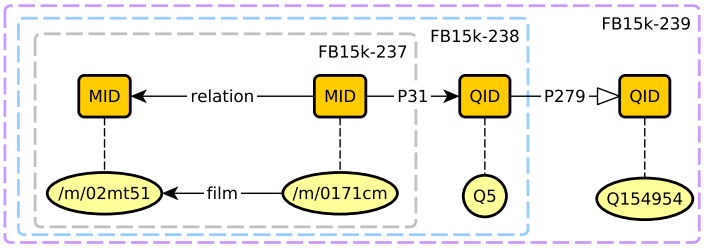

Fig. 1: A graphical representation of the types of triples contained in each of the datasets. The yellow ellipses are a set of triples extracted from T_{239}. The dashed boxes correspond to the colors in Figure 2.

and validating) for the new datasets compared to the FB15k and FB15k-237. Directions accessing these datasets, the code, and further documentation, including licensing, are reported in Section 3.3. Section 3.4 shows our evaluation strategy.

3.1 Creating FB15k-238

FB15k-238 is an augmentation of the FB15k-237 dataset: it includes exactly one new relation, P31 (hence the 238). P31 is taken from Wikidata and has the label "instanceOf" or "isA."[1] To construct this augmented dataset, we looped through each Freebase entity (MID) and queried Wikidata for its type via the P31 property.

We note that not every MID remains incorporated into Wikidata from the original transfer (either they were never transferred or, over time, were for some reason removed).[2] As such, our FB15k-238 dataset is missing the type information for 42 entities.

3.2 Creating FB15k-239

FB15k-239 is an augmentation of FB15k-238: it includes exactly one new relation, P239 (hence the 239). P279 is taken from Wikidata and hase the label subclass of.[3] To construct this augmented dataset, for each Wikidata entity added in FB15k-238, we queried Wikidata for its superclass via the P279 property.

[1] https://www.wikidata.org/wiki/Property:P31

[2] When querying with Freebase MID /m/01sxq9, Wikidata had once listed this entity as "Bebe Neuwirth" but has since removed the MID Property from the respective page. Google remarks during the data migration, they were faced with technical and non-technical challenges, which resulted in the missing MIDs on Wikidata [21].

[3] https://www.wikidata.org/wiki/Property:P279

Dataset	# Entities	# Rel.s	# Train	# Validation	#Test
FB15k-237	14541	237	272115	17535	20466
FB15k-238	16414	238	293471	31482	35257
FB15k-239	17494	239	296822	33879	37738

Table 2: This table shows a comparison of different counts for our Freebase datasets and their augmentations.

3.3 Availability & Licensing

The trained models are provided through a Zenodo repository. [4]The datasets and code for constructing these datasets is also provided online via a GitHub repository.[5]These resources are licensed permissibly under the MIT License.

3.4 Evaluation

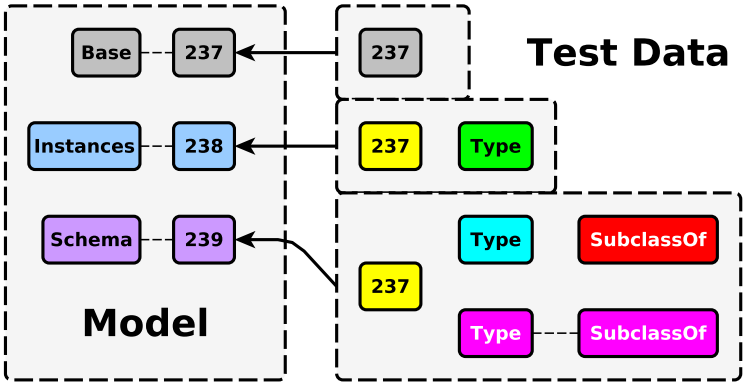

Fig. 2: This figure provides a graphical overview of the different KGE models and their corresponding augmentations (i.e., FB15k-238 contains the instanceOf properties). The right hand side shows the different sets of test data used to evaluate the models.

In essence, we can consider our evaluation strategy to constitute a larger ablation study, where we are adding and removing components of the training and evaluation data to measure isolated performance, that is the *impact* of different graph structures have on overall KGE performance. We define nine evaluation components, which are to be executed against the six models implemented in DGL-KE. These 54 results are reported in the next section.

[4] Available upon request during blind reviewing.
[5] `https://anonymous.4open.science/r/kge-impact-22C5/`

For convenience, we define notation for our evaluation components. Let $\mathbf{M}_x$ be the set of KGE Models which results from the training over the dataset FB15k-x. Specifically, M_{237} is the set of models trained over FB15k-237. This is our base case, and it is described below. M_{238} is the set of models trained over when the type information (or instance metadata) is present. Finally, M_{239} is the set of models resulting from training over the datasets where there is both the type information and a basic subsumption hierarchy for those types. Test data, which we consider to be a set, is represented by T_x. We use the arrow ($\leftarrow$) to indicate which test data T_x is being fed to the model. An evaluation for some input test data is executed against each of the six models. For example, our base case, is replicating the results of FB15k-237 training data on each of the models, is stated as $M_{237} \leftarrow T_{237}$. This is represented by the gray boxes in Figure 2. It was necessary to do this replication step to ensure we were fairly comparing all models for a given machine configuration.

The **yellow boxes** correspond to $M_{238} \leftarrow T_{237}$ and $M_{239} \leftarrow T_{237}$. The purpose of these evaluations is to test if *training* with increased metadata (i.e., type information and a basic subsumption hierarchy) would improve the results, when not evaluating performance on the type assertion triples.

The **green box** corresponds to $M_{238} \leftarrow T_{238} - T_{237}$. This evaluation tests if performance on specifically type prediction differs from overall performance. Analogously, the **cyan box** is represented by $M_{239} \leftarrow T_{238} - T_{237}$. This evaluation is performed to test if training when the type information augmented with the subsumption hierarchy improves type prediction.

The **red box** corresponds to $M_{239} \leftarrow T_{239} - T_{238}$. In this case, we wish to test if the model performs well on solely (basic) ontology prediction tasks. That is, whether or not subsumption relations are correctly predicted.

The linked **fuchsia boxes** correspond to $M_{239} \leftarrow T_{239} - T_{237}$. This evaluation is to test for isolated results on the metadata aspects, but trained with the base facts.

3.5 Evaluation Metrics

Mean Rank (MR) [2] is a statistical metric representing the average position or ordinal rank assigned to a set of items in a given ranking. A lower score of Mean Rank indicates a better performing model.

Mean Reciprocal Rank (MRR) [2] is a statistical measure that assesses the average of the reciprocals of the ranks assigned to relevant items in a ranked list. A higher MRR score, constrained to {0,1}, indicates a better performing model.

Hits@K [2] is an evaluation metric that measures the number of relevant items present in the top-k positions of a ranked list. A higher value indicates a better performing model. Our evaluation uses k at 1, 3, and 10.

Model	Metrics	FB15k-237	FB15k-238	$\Delta_{237-238}$	FB15k-239	$\Delta_{237-239}$
TransE	MRR	0.4143	0.7219	-0.3076	**0.7342**	-0.3199
	MR	33.9947	11.1185	22.8762	**10.3057**	23.6890
	HITS@1	0.2982	0.6440	-0.3458	**0.6569**	-0.3587
	HITS@3	0.4701	0.7729	-0.3028	**0.7836**	-0.3134
	HITS@10	0.6394	0.8577	-0.2183	**0.8714**	-0.2320
TransR	MRR	**0.2901**	0.2019	0.0882	0.2361	0.0540
	MR	**152.6647**	241.1533	-88.4886	209.9060	-57.2413
	HITS@1	**0.2247**	0.1538	0.0709	0.1832	0.0414
	HITS@3	**0.3148**	0.2153	0.0995	0.2526	0.0622
	HITS@10	**0.4066**	0.2871	0.1195	0.3292	0.0774
ComplEx	MRR	**0.3064**	0.1296	0.1768	0.1909	0.1155
	MR	**84.6207**	225.5332	-140.9126	129.6837	-45.0630
	HITS@1	**0.2034**	0.0809	0.1225	0.1152	0.0881
	HITS@3	**0.3532**	0.1331	0.2201	0.2053	0.1479
	HITS@10	**0.5001**	0.2206	0.2795	0.3420	0.1581
DistMult	MRR	**0.3213**	0.1644	0.1569	0.2344	0.0869
	MR	**80.7576**	137.8459	-57.0883	108.8363	-28.0787
	HITS@1	**0.2180**	0.0865	0.1315	0.1452	0.0728
	HITS@3	**0.3672**	0.1785	0.1887	0.2601	0.1071
	HITS@10	**0.5176**	0.3216	0.1959	0.4122	0.1054
RESCAL	MRR	0.3520	0.3132	0.0388	**0.3939**	-0.0419
	MR	126.5442	134.8030	-8.2588	**104.7880**	21.7562
	HITS@1	0.2766	0.2478	0.0287	**0.3142**	-0.0376
	HITS@3	0.3884	0.3366	0.0517	**0.4305**	-0.0421
	HITS@10	0.4789	0.4320	0.0469	**0.5388**	-0.0599
RotatE	MRR	**0.0769**	0.0751	0.0018	0.0629	0.0140
	MR	**277.2960**	287.1963	-9.9003	298.5854	-21.2894
	HITS@1	**0.0419**	0.0394	0.0025	0.0344	0.0075
	HITS@3	**0.0757**	0.0728	0.0028	0.0592	0.0165
	HITS@10	0.1331	**0.1361**	-0.0029	0.1068	0.0264

Table 3: This table reports the results of evaluating each of the models against their respective training data.

Model	Metrics	FB15k-237	FB15k-238	$\Delta_{237-238}$	FB15k-239	$\Delta_{237-239}$
TransE	MRR	0.4143	0.5489	-0.1346	**0.5566**	-0.1423
	MR	33.9947	17.2292	16.7655	**16.1160**	17.8787
	HITS@1	0.2982	0.4349	-0.1367	**0.4440**	-0.1458
	HITS@3	0.4701	0.6152	-0.1450	**0.6195**	-0.1494
	HITS@10	0.6394	0.7575	-0.1180	**0.7668**	-0.1274
TransR	MRR	0.2901	0.3037	-0.0136	**0.3058**	-0.0157
	MR	152.6647	138.4772	14.1875	**128.6830**	23.9817
	HITS@1	0.2247	0.2401	-0.0155	**0.2404**	-0.0158
	HITS@3	0.3148	0.3266	-0.0118	**0.3293**	-0.0145
	HITS@10	0.4066	0.4188	-0.0123	**0.4226**	-0.0160
ComplEx	MRR	**0.3064**	0.0840	0.2224	0.1297	0.1767
	MR	**84.6207**	281.7108	-197.0901	179.7169	-95.0962
	HITS@1	**0.2034**	0.0425	0.1609	0.0652	0.1382
	HITS@3	**0.3532**	0.0859	0.2673	0.1398	0.2134
	HITS@10	**0.5001**	0.1612	0.3389	0.2565	0.2436
DistMult	MRR	**0.3213**	0.1097	0.2116	0.1537	0.1676
	MR	**80.7576**	190.6282	-109.8706	158.1567	-77.3991
	HITS@1	**0.2180**	0.0508	0.1671	0.0800	0.1380
	HITS@3	**0.3672**	0.1147	0.2525	0.1696	0.1976
	HITS@10	**0.5176**	0.2235	0.2941	0.2993	0.2183
RESCAL	MRR	0.3520	**0.3771**	-0.0251	0.3766	-0.0247
	MR	126.5442	119.2274	7.3168	**116.6290**	9.9152
	HITS@1	0.2766	**0.3108**	-0.0342	0.3055	-0.0290
	HITS@3	0.3884	0.4043	-0.0160	**0.4076**	-0.0192
	HITS@10	0.4789	0.4957	-0.0168	**0.5022**	-0.0233
RotatE	MRR	0.0769	**0.0775**	-0.0006	0.0664	0.0104
	MR	277.2960	265.2055	12.0905	**257.6581**	19.6379
	HITS@1	**0.0419**	0.0408	0.0011	0.0341	0.0078
	HITS@3	0.0757	**0.0767**	-0.0010	0.0625	0.0131
	HITS@10	0.1331	**0.1387**	-0.0055	0.1189	0.0143

Table 4: This table reports the results of testing each of the models against solely the FB15k-237 training data (i.e., T_{237}).

Model	Metrics	$M_{238} \leftarrow$	$M_{239} \leftarrow$		
		$T_{238-237}$	$T_{238-237}$	$T_{239-238}$	$T_{239-237}$
TransE	MRR	0.9590	0.9465	0.9495	0.9470
	MR	2.6546	3.1323	6.0362	3.4810
	HITS@1	0.9280	0.9122	0.9132	0.9128
	HITS@3	0.9910	0.9785	0.9851	0.9799
	HITS@10	0.9970	0.9958	0.9910	0.9955
TransR	MRR	0.0632	0.1643	0.0770	0.1522
	MR	383.1885	254.3162	618.0510	306.3611
	HITS@1	0.0377	0.1215	0.0611	0.1139
	HITS@3	0.0645	0.1744	0.0832	0.1617
	HITS@10	0.1034	0.2377	0.1002	0.2175
ComplEx	MRR	0.1904	0.2250	0.4818	0.2614
	MR	147.8925	75.8214	38.9972	70.3205
	HITS@1	0.1317	0.1347	0.3841	0.1702
	HITS@3	0.1957	0.2433	0.5261	0.2839
	HITS@10	0.3033	0.4047	0.6724	0.4436
DistMult	MRR	0.2399	0.2989	0.5142	0.3316
	MR	64.7679	53.2142	35.7648	50.5354
	HITS@1	0.1350	0.1896	0.4160	0.2245
	HITS@3	0.2652	0.3329	0.5661	0.3685
	HITS@10	0.4593	0.5212	0.6941	0.5479
RESCAL	MRR	0.2238	0.3830	0.5886	0.4127
	MR	156.3768	89.4667	99.9538	90.8974
	HITS@1	0.1583	0.2852	0.5412	0.3223
	HITS@3	0.2437	0.4319	0.6087	0.4571
	HITS@10	0.3440	0.5639	0.6764	0.5801
RotatE	MRR	0.0707	0.0679	0.0049	0.0584
	MR	317.6899	293.1596	669.9831	347.1890
	HITS@1	0.0364	0.0418	0.0012	0.0352
	HITS@3	0.0656	0.0645	0.0020	0.0549
	HITS@10	0.1329	0.1071	0.0068	0.0923

Table 5: This table reports the results of our ablation-like study, where we change which component of the data against which we evaluate.

4 Results

Our results are reported in Tables 3–5. They are segmented into, broadly, three target goals – measuring holistic performance, measuring performance solely on FB15k-237 test data, and an ablation-like study for different data components. Results are discussed in Section 5.

4.1 Holistic Performance

These results are reported in Table 3. We bold the top performing results; the second best results are underlined. The fifth and seventh columns, separated by double vertical lines, report the differences in performance against the base result. The difference Δ_{x-y} corresponds to the difference of the results against M_x minus the results against M_y. This means that negative values indicate an improvement for the MRR and Hits@K metrics, and regression for MR metric. Discussion of these results appears in Section 5.1.

4.2 Performance on Base Data

These results are reported in Table 4. As before stated, these results correspond to the yellow boxes in Figure 2. The table is organized and depicted in the same manner as Table 3. Discussion of these results appears in Section 5.2.

4.3 Ablation Study

These results are reported in Table 5. Columns are labeled with a shortened notation from Section 3.4, where T_{x-y} indicates the set difference of training data for the different datasets. These results are not compared against each other, graphically, as they are not all measuring the same thing. Discussion of these results appears in Section 5.3.

5 Discussion

The overarching purpose of this study is to identify how changing the structure of the graph impacts the performance of different KGE models. Specifically, we haven chosen to make changes to the graph structure based on how semantic and symbolic metadata is added to the graph. Specifically, this means we have generated dataset augmentations:

- for each entity in FB15k-237, we have type information (P31 and associated entities) to produce FB15k-238 and
- for each type added, we have its superclass (mimicking a shallow subsumption hierarchy) to produce FB15k-239.

This process is repeated for each set of triples (i.e., training, validation, and test).

First, we wish to assess the comprehensive, or holistic, performance. That is, measuring the performance against our metrics over the same type of data the models were trained. In our second stage, we remove the semantic information to see if performance strictly on the factual (i.e., base) data is improved. Finally, in our ablation-like study, we inspect how these models perform strictly over the semantic information.

5.1 Holistic Performance

The holistic performance is reported in Table 4.1. We can see right away that the addition of semantic metadata has an impact on the overall performance of the KGE models. In particular, TransE performs exceptionally better in both cases (T_{238} and T_{239}) and across all metrics as the semantic information is added. TransE likely does well, as many new relationships were added of the form $h_{i...n} \xrightarrow{r} t$, of which TransE excels in modeling. This is likely also reflected by the marginal improvement for $\Delta_{238-239}$, as fewer new triples were added, but generally had the same format. This likely had a stronger clustering effect, for which we should see evidence in the next stage. Conversely, TransR likely does poorly for the same reason TransE improves.

DistMult and ComplEx exhibit the same overall behavior to varying degrees. Adding semantic information, at all, negatively impacts performance, but, curiously, the inclusion of *all* semantic information seems to marginally improve performance. This is likely due to the fact that DistMult is not capable of representing antisymmetric relationships (such as P31). Yet, ComplEx, which should, does poorly across all data sets.

RESCAL performs similarly to TransE, but its worse performance is when the type information is included. It does perform better than DistMult, as expected, given DistMult's restrictions.

Finally, RotatE performs progressively worse as semantic information is added, and, in general, performs extremely poorly.

5.2 Performance on Base Data

The purpose of this study was to determine if *training* over semantically augmented data improves performance of the KGE models. To determine this, we evaluate the models with strictly the training data from FB15k-237 (T_{237}). This also has the benefit of helping determine the origin of the impact on the performance results. That is, are improvements driven by "easy-to-guess" triples that are added to the evaluation, or is performance improved across the board? These results are reported in Table 4.

Indeed, we do see that performance is improved without the semantic information as part of the evaluation. However, the improvement is much more modest, indicating that there is an element of truth to TransE being capable of easily predicting the types of entities. However, we also hypothesize that during training, the presence of the typing triples provided a clustering impetus allowing for a consistent translation.

TransR also indicates a marginal improvement, but a consistent one, and enough to show that M_{239} is the most performant.

On the other hand, DistMult and ComplEx have the same relative performance as in the previous stage. RESCAL and, in particular, RotatE perform poorly. This indicates that the presence of semantic information during training hinders the ability to learn other sorts of relations.

5.3 Ablation Study

The final stage of this study is an ablation-like study, where we add and remove different components of the data in-order to understand the isolated performance on those types of triples. This corresponds to Table 5. Essentially, these evaluation components are type prediction tasks. Specifically, we test for the type of a particular entity ($T_{238-237}$), subsumption ($T_{239-238}$), and both sets combined against models trained against that data.

TransE does surprisingly well, having nearly perfect performance. This may also be well to the hypothesized strong clustering effect, making simple connection-based predictions from clearly defined spaces.

TransR and RotatE have the worst performance, where RESCAL, DistMult, and ComplEx – all belonging to the same family – still perform quite poorly, but seem to do at least middling well over (specifically) the subsumption prediction task.

What remains to be answered are whether or not the poor performance on the more complex KGE models, and especially the non-translational models, arises due to a relatively frequent appearance of specific relations resulting competing with infrequently appearing relations. Consequently, this may result in no clear differentiation between types of nodes.

6 Conclusion

Knowledge graphs are an increasingly important resource in both academia and industry. Many new methods for utilizing them have surfaced over the years [3]. On the other hand, we have a plethora of methodologies for producing KGs (and ontologies). To the author's knowledge, there is no systematic review of how the different KGs – and in particular their frequently dissimilar triple patterns (i.e., graph structure), differently impact the performance of downstream tasks (e.g., knowledge graph embeddings). The purpose of this work is to begin this review.

Concretely, we have constructed two new datasets, from the oft-used FB15k-237, which reflect additional equipping of semantics. These are FB15k-238 and FB15k-239, which – respectively – include type information (i.e., instanceOf – P31) and subclass information (i.e., subclass of – P279) and their associated entities. We then measured the changes in performance when including these additional triples during training. We then conducted additional exploration to determine a more fine-grained understanding of the performance of the model when isolating the base ("factual") data and semantic data (metadata).

There is significant opportunity for further work in this space, as well as ample dataset options for repeating this study to get a broader overview. In particular, we see the following immediate next steps:

1. Preliminary visualizations show that there are inherent clusters to FB15k-237 without semantic information, but these clusters seem to fray (i.e., lose crispness) after

training with semantic data. One exploration would be to identify more strictly how these clusters degrade. On the other hand, in the TransE model there is likely to be some sort of strong clustering effect that can be leveraged as an additional source of mining.
2. Further analysis of the FB15k-23x datasets to understand sampling effects, given the addition of new triples, and measure any impact on the performance.
3. Repeating this process of additional datasets. Identifying type and subclass information was straightforward due to the inherent overlap between Freebase IDs and Wikidata IDs. Other common benchmarks are WN18 [7] and YAGO [30] datasets.
4. Increasing the breadth of semantic richness and other schema level manipulations. For now, we have only explored extremely shallow additions of type information and one-hop subsumption of classes. How is performance impacted when we have a full OWL [12] schema with reified nodes?
5. Finally, incorporating [10] into our workflow – which was unfortunately discovered too late in our experiment to properly leverage.

Acknowledgement. This work was, in part, funded by the National Science Foundation under Grant 2333532; Proto-OKN Theme 3: An Education Gateway for the Proto-OKN. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

1. An overview of the common core ontologies (2019), https://api.semanticscholar.org/CorpusID:199556341
2. Akrami, F., Saeef, M.S., Zhang, Q., Hu, W., Li, C.: Realistic re-evaluation of knowledge graph completion methods: An experimental study (2020)
3. Ali, M., Berrendorf, M., Hoyt, C.T., Vermue, L., Galkin, M., Sharifzadeh, S., Fischer, A., Tresp, V., Lehmann, J.: Bringing light into the dark: A large-scale evaluation of knowledge graph embedding models under a unified framework. IEEE Trans. Pattern Anal. Mach. Intell. **44**(12), 8825–8845 (2022). https://doi.org/10.1109/TPAMI.2021.3124805, https://doi.org/10.1109/TPAMI.2021.3124805
4. Bebee, B.R., Choi, D., Gupta, A., Gutmans, A., Khandelwal, A., Kiran, Y., Mallidi, S., McGaughy, B., Personick, M., Rajan, K.J., Rondelli, S., Ryazanov, A., Schmidt, M., Sengupta, K., Thompson, B.B., Vaidya, D., Wang, S.X.: Amazon neptune: Graph data management in the cloud. In: International Workshop on the Semantic Web (2018), https://api.semanticscholar.org/CorpusID:52977077
5. Bezerra, C., Freitas, F., Santana da Silva, F.: Evaluating ontologies with competency questions. pp. 284–285 (11 2013). https://doi.org/10.1109/WI-IAT.2013.199
6. Blomqvist, E., Hammar, K., Presutti, V.: Engineering Ontologies with Patterns – The eXtreme Design Methodology. In: Hitzler, P., Gangemi, A., Janowicz, K., Krisnadhi, A., Presutti, V. (eds.) Ontology Engineering with Ontology Design Patterns – Foundations and Applications, Studies on the Semantic Web, vol. 25, pp. 23–50. IOS Press (2016)
7. Bordes, A., Usunier, N., Garcia-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2. p. 2787–2795. NIPS'13, Curran Associates Inc., Red Hook, NY, USA (2013)
8. Dave, B., Shimizu, C.: Towards understanding the impact of schema on knowledge graph embeddings (invited) (2023), in press
9. Fernandez-Lopez, M., Gomez-Perez, A., Juristo, N.: Methontology: from ontological art towards ontological engineering. In: Proceedings of the AAAI97 Spring Symposium. pp. 33–40 (March 1997)
10. Heist, N., Hertling, S., Paulheim, H.: Kgreat: A framework to evaluate knowledge graphs via downstream tasks. In: Frommholz, I., Hopfgartner, F., Lee, M., Oakes, M., Lalmas, M., Zhang, M., Santos, R.L.T. (eds.) Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023. pp. 3938–3942. ACM (2023). https://doi.org/10.1145/3583780.3615241, https://doi.org/10.1145/3583780.3615241
11. Hitzler, P.: Semantic Web: A review of the field. Communications of the ACM (2021), to appear
12. Hitzler, P., Parsia, B., Rudolph, S., Patel-Schneider, P., Krötzsch, M.: OWL 2 web ontology language primer (second edition). W3C recommendation, W3C (Dec 2012), https://www.w3.org/TR/2012/REC-owl2-primer-20121211/
13. Hitzler, P., Rayan, R., Zalewski, J., Norouzi, S.S., Eberhart, A., Vasserman, E.Y.: Deep deductive reasoning is a hard deep learning problem (2023), under review
14. Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., de Melo, G., Gutierrez, C., Kirrane, S., Gayo, J.E.L., Navigli, R., Neumaier, S., Ngomo, A.N., Polleres, A., Rashid, S.M., Rula, A., Schmelzeisen, L., Sequeda, J.F., Staab, S., Zimmermann, A.: Knowledge graphs. ACM Comput. Surv. **54**(4), 71:1–71:37 (2022). https://doi.org/10.1145/3447772, https://doi.org/10.1145/3447772
15. Iferroudjene, M., Charpenay, V., Zimmermann, A.: FB15k-CVT: A Challenging Dataset for Knowledge Graph Embedding Models. In: NeSy 2023, 17th International Workshop on

Neural-Symbolic Learning and Reasoning. pp. 381–394. Siena, Italy (Jul 2023), https://hal-emse.ccsd.cnrs.fr/emse-04081543
16. Kejriwal, M., Knoblock, C., Szekely, P.: Knowledge Graphs: Fundamentals, Techniques, and Applications. Adaptive Computation and Machine Learning series, MIT Press (2021), https://books.google.com/books?id=iqvuDwAAQBAJ
17. Kulvatunyou, B., Wallace, E.K., Kiritsis, D., Smith, B., Will, C.: The industrial ontologies foundry proof-of-concept project. In: Advances in Production Management Systems (2018), https://api.semanticscholar.org/CorpusID:52110800
18. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. Proceedings of the AAAI Conference on Artificial Intelligence **29**(1) (Feb 2015). https://doi.org/10.1609/aaai.v29i1.9491, https://ojs.aaai.org/index.php/AAAI/article/view/9491
19. Nickel, M., Tresp, V., Kriegel, H.P.: A three-way model for collective learning on multi-relational data. In: Proceedings of the 28th International Conference on International Conference on Machine Learning. p. 809–816. ICML'11, Omnipress, Madison, WI, USA (2011)
20. Noy, N.F., Gao, Y., Jain, A., Narayanan, A., Patterson, A., Taylor, J.: Industry-scale knowledge graphs: lessons and challenges. Commun. ACM **62**(8), 36–43 (2019). https://doi.org/10.1145/3331166, https://doi.org/10.1145/3331166
21. Pellissier Tanon, T., Vrandečić, D., Schaffert, S., Steiner, T., Pintscher, L.: From freebase to wikidata: The great migration. In: Proceedings of the 25th International Conference on World Wide Web. p. 1419–1428. WWW '16, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE (2016). https://doi.org/10.1145/2872427.2874809, https://doi.org/10.1145/2872427.2874809
22. Rdf-star working group, https://www.w3.org/groups/wg/rdf-star/
23. Rossi, A., Barbosa, D., Firmani, D., Matinata, A., Merialdo, P.: Knowledge graph embedding for link prediction: A comparative analysis. ACM Trans. Knowl. Discov. Data **15**(2) (jan 2021). https://doi.org/10.1145/3424672, https://doi.org/10.1145/3424672
24. Saeedi, A., Peukert, E., Rahm, E.: Using Link Features for Entity Clustering in Knowledge Graphs, pp. 576–592 (06 2018). https://doi.org/10.1007/978-3-319-93417-4_37
25. Shimizu, C., Hammar, K., Hitzler, P.: Modular ontology modeling. Semantic Web **14**(3), 459–489 (2023), https://doi.org/10.3233/SW-222886
26. Smith, B.: The basic tools of formal ontology. In: Formal Ontology in Information Systems (1998)
27. Suárez-Figueroa, M.C., Gómez-Pérez, A., Fernández-López, M.: The NeOn Methodology for Ontology Engineering, pp. 9–34. Springer Berlin Heidelberg, Berlin, Heidelberg (2012). https://doi.org/10.1007/978-3-642-24794-1_2, https://doi.org/10.1007/978-3-642-24794-1_2
28. Sun, Z., Deng, Z., Nie, J., Tang, J.: Rotate: Knowledge graph embedding by relational rotation in complex space. CoRR **abs/1902.10197** (2019), http://arxiv.org/abs/1902.10197
29. Tan, Z., Zhao, X., Fang, Y., Ge, B., Xiao, W., Gomez-Pulido, J.A.: Knowledge graph representation via similarity-based embedding. Sci. Program. **2018** (jan 2018). https://doi.org/10.1155/2018/6325635, https://doi.org/10.1155/2018/6325635
30. Thomas Pellissier Tanon, Gerhard Weikum, F.M.S.: Yago 4: A reason-able knowledge base (2020). https://doi.org/10.1007/978-3-030-49461-2_34
31. Toutanova, K., Chen, D.: Observed versus latent features for knowledge base and text inference (07 2015). https://doi.org/10.18653/v1/W15-4007
32. Trouillon, T., Welbl, J., Riedel, S., Éric Gaussier, Bouchard, G.: Complex embeddings for simple link prediction (2016)

33. Webber, J.: A programmatic introduction to neo4j. In: Leavens, G.T. (ed.) SPLASH'12 - Proceedings of the 2012 ACM Conference on Systems, Programming, and Applications: Software for Humanity, Tucson, AZ, USA, October 21-25, 2012. pp. 217–218. ACM (2012). https://doi.org/10.1145/2384716.2384777, https://doi.org/10.1145/2384716.2384777
34. Yang, B., tau Yih, W., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases (2015)
35. York Sure, S.S., Studer, R.: On-to-knowledge methodology (OTKM), pp. 117–132 (2004). https://doi.org/https://doi.org/10.1007/978-3-540-24750-0_6
36. Zheng, D., Song, X., Ma, C., Tan, Z., Ye, Z., Dong, J., Xiong, H., Zhang, Z., Karypis, G.: Dgl-ke: Training knowledge graph embeddings at scale. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 739–748. SIGIR '20, Association for Computing Machinery, New York, NY, USA (2020)