

Constrain the Relational Dimensions with Conjugation for Complex Knowledge Graph Embeddings

Xincan Feng, Yuchang Cheng,
Taro Watanabe, Nobuhiro Yugami

empirical analysis: learning curve? train data, 5-tests, mrr, dataset size (big dataset compare) dataset reduce

1 Abstract

Knowledge graph embedding (KGE) models have been a popular approach to do link prediction task for knowledge graph completion (KGC). KGE models first choose a representation method, then employ a transformation function to map nodes via edges into a corresponding vector space in order to measure the likelihood of the links. While mapping the individual nodes, the structure of the subgraphs is also transformed. In order to make the KGE models more expressive, researchers are using more and more sophisticated representations, spaces and transformation functions.

However, as those become more complicated, it becomes harder for researchers to distinguish the redundancy of their models. Our research proposed a dimension restriction method to reduce the redundancy in complex KGE models. Our research has verified and demonstrated the relational nature of dimensions in two best KGE models: 5★E (Nayyeri et al. 2021) and ComplEx (Trouillon et al. 2016). Specifically, we proposed conjugate method to add restriction in the dimensions of parameters matrix, so that to reduce the calculation while receiving comparable performance.

2 Introduction

In recent years, many models have focused on different expressions and different spatial curvatures, which all imply one point: there are interactions between dimensions of KGE models.

Knowledge graphs (KGs) with their graph-based knowledge representation in the form of (head, relation, tail) triples, have become a leading technology of recent years in AI-based tasks including question answering, data integration, and recommender systems (Ji et al. 2020). However, KGs are incomplete, and the application consuming them is affected by this problem. Knowledge graph embedding (KGE) is a prominent approach used for knowledge graph completion (KGC) by predicting missing links.

Every KGE model defines a specific representation, and uses a transformation function to map entities (nodes) of the KG through relations into a corresponding vector space, and then calculate the plausibility of triples via a corresponding score function. Thus, there are three main directions in improving the KGE models: representations, vector spaces, and transformation functions. **We can generally conclude that:**

- (i) Compared with real number representation, the composition of complex number can handle a larger variety of binary relations, among them symmetric and antisymmetric relations (Trouillon et al. 2016).
- (ii) Compared with Euclidean space, the Hyperbolic space models can save more structures using fixed or trainable curvatures in different positions for hierarchical relations (Chami et al. 2020).
- (iii) Compared with naive addition or multiplication transformation functions, more specific designed addition, multiplication and their composition, including projective geometric functions, support multiple simultaneous transformations, and are more capable to represent multiple structures in the multi-relational knowledge graphs.

The transformation functions distinguish the extent to which a KGE model is able to learn complicated motifs and patterns formed by combinations of the nodes and edges. Representation methods of different spaces are where all transformation functions first rest on. Dimensions are the base of the representation methods and thus determine the transformation functions' capabilities.

Our intuitions

- (i) The increased performance from real number representation to complex number representation, which enables the imaginary and real part dimensions to be relational, indicates the **hidden relations** between parameter dimensions.
- (ii) The increased performance from Euclidean space to Hyperbolic space, which enables the distances or angles of **different positions** to vary in different degrees, indicates different positions in the parameter dimensions have different degrees of freedom. Or rather, the fixed or trainable curvatures in Hyperbolic embeddings can learn uneven information to affect the link plausibility result. We hypothesis that the uneven parameter dimensions is capable to do that too.

The **"degree of freedom" or call it "restriction" methods on dimensions**, are similar to **"attention" or "weight" methods on parameters** (Vaswani et al. 2017). They all explore the "relationship problem" among the feature parameters of the data.

Nonetheless, the "restriction" on dimensions wins in that it goes deeper to the root of the "relationship", thus is possible to be more effective by reducing the calculation. Concretely speaking, the same parameters in different positions of transformation functions, for many KGE models, are adequate to express the model capabilities.

3 Related work

KGE models that are classified according to their representation methods:

Real embeddings Translation approaches include TransE (Bordes et al. 2013) and its variants (Ji et al. 2015, Lin et al. 2015). Although these models are fairly simple and have few parameters, they fail to encode important logical properties (e.g., translations can't encode symmetry).

Complex embeddings ComplEx (Trouillon et al. 2016) gives a clear comparison with respect to existing approaches using only real numbers by presenting an equivalent reformulation of their model that involves only real embeddings. RotatE (Sun et al. 2019) does require additional optimization component to effectively draw negative samples for training. 5★E (Nayyeri et al. 2021) uses Möbius transformation as the transformation function, which is the composition of four functions that can represent five transformations.

KGE models that are classified according to their vector spaces:

Euclidean embeddings Tensor factorization methods such as RESCAL (Nickel et al. 2011) and DistMult (Yang et al. 2015) are designed based on element-wise multiplication of transformed head and tail. In this case, the plausibility

of triples is measured based on the angle of transformed head and tail.

Hyperbolic embeddings MuRP (Balazevic et al. 2019) minimizes hyperbolic distances between a re-scaled version of the head entity embedding and a translation of the tail entity embedding. It uses hyperbolic embeddings with fewer dimensions than its Euclidean analogues. ATTH (Chami et al. 2020) leverages trainable hyperbolic curvatures per relationship to simultaneously capture logical patterns and hierarchies. However, it also needs optimization methods in tangent space (i.e., Euclidean).

4 Method

Our hypothesis is that: The transformation functions of complex embeddings are possibly to be improved in efficiency by conjugation methods.

Complex embedding performance has already shown improvement compared to real embeddings by setting relational restrictions for the transformation function parameters. Our goal is to find out if there exists redundancy in the transformation function capabilities.

i denotes the imaginary number, $a_1, a_2, \dots, d_1, d_2$ are the real parameters in the transformation function, $Re(x)$ is the real part of the complex number x , \bar{x} is the complex conjugate of x .

5★E:

Transformation function:

$$a = a_1 + a_2i, b = b_1 + b_2i, c = c_1 + c_2i, d = d_1 + d_2i$$

$$\begin{aligned} x_1 + x_2i &\rightarrow \begin{bmatrix} x_1 + x_2i \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} a_1 + a_2i & b_1 + b_2i \\ c_1 + c_2i & d_1 + d_2i \end{bmatrix} \begin{bmatrix} x_1 + x_2i \\ 1 \end{bmatrix} \rightarrow \frac{(a_1 + a_2i)(x_1 + x_2i) + b_1 + b_2i}{(c_1 + c_2i)(x_1 + x_2i) + d_1 + d_2i} \\ &= \frac{a_1x_1 + a_1x_2i + a_2x_1i - a_2x_2 + b_1 + b_2i}{c_1x_1 + c_1x_2i + c_2x_1i - c_2x_2 + d_1 + d_2i} \end{aligned}$$

Score function:

$$f(h, r, t) = Re(\langle h_r, t \rangle) = Re\left(\frac{h_r \cdot t}{\sqrt{h_r^2} \cdot \sqrt{t^2}}\right)$$

Regularization: N3 regularization

Theoretical analysis:

Möbius transformation ϑ combines five subsequent transformations $\vartheta_1, \vartheta_2, \vartheta_3, \vartheta_4$. Thus is capable of performing five kinds of transformations simultaneously.

$$\vartheta = \vartheta_4 \circ \vartheta_3 \circ \vartheta_2 \circ \vartheta_1$$

translation:

$$\vartheta_1 = x_1 + x_2i + \frac{d_1 + d_2i}{c_1 + c_2i}$$

inversion and reflection w.r.t. real axis:

$$\vartheta_2 = \frac{1}{x_1 + x_2i}$$

homothety and rotation:

$$\vartheta_3 = \frac{(b_1 + b_2i)(c_1 + c_2i) - (a_1 + a_2i)(d_1 + d_2i)}{(c_1 + c_2i)^2} (x_1 + x_2i)$$

translation:

$$\vartheta_4 = x_1 + x_2i + \frac{a_1 + a_2i}{c_1 + c_2i}$$

5★E_{conj}:

$c = \bar{b}, d = \bar{a}$, thus $a = a_1 + a_2i, b = b_1 + b_2i, c = b_1 - b_2i, d = a_1 - a_2i$

($c = -\bar{b}, d = \bar{a}$ works fine as well, and this forms a unitary Möbius transformation.)

$$\begin{aligned} x_1 + x_2i &\rightarrow \begin{bmatrix} x_1 + x_2i \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} a_1 + a_2i & b_1 + b_2i \\ b_1 - b_2i & a_1 - a_2i \end{bmatrix} \begin{bmatrix} x_1 + x_2i \\ 1 \end{bmatrix} \rightarrow \frac{(a_1 + a_2i)(x_1 + x_2i) + b_1 + b_2i}{(b_1 - b_2i)(x_1 + x_2i) + a_1 - a_2i} \\ &= \frac{a_1x_1 + a_1x_2i + a_2x_1i - a_2x_2 + b_1 + b_2i}{b_1x_1 + b_1x_2i - b_2x_1i + b_2x_2 + a_1 - a_2i} \end{aligned}$$

Score function:

$$f(h, r, t) = \text{Re}(\langle h_r, t \rangle)$$

Regularization: N3 regularization

Theoretical analysis:

$$\vartheta = \vartheta_4 \circ \vartheta_3 \circ \vartheta_2 \circ \vartheta_1$$

translation:

$$\vartheta_1 = x_1 + x_2i + \frac{a_1 - a_2i}{b_1 - b_2i}$$

inversion and reflection w.r.t. real axis:

$$\vartheta_2 = \frac{1}{x_1 + x_2i}$$

homothety and rotation:

$$\vartheta_3 = \frac{(b_1 + b_2i)(b_1 - b_2i) - (a_1 + a_2i)(a_1 - a_2i)}{(b_1 - b_2i)^2} (x_1 + x_2i) = \frac{b_1^2 + b_2^2 - a_1^2 - a_2^2}{(b_1 - b_2i)^2} (x_1 + x_2i)$$

translation:

$$\vartheta_4 = x_1 + x_2i + \frac{a_1 + a_2i}{b_1 - b_2i}$$

5 Experiments and discussion

5.1 Experimental Setup

Metrics We followed the standard evaluation protocol for KGE models as 5★E (Nayyeri et al. 2021). T : the rank set of truth, r_i : the rank position r of the first true entity for the i -th query. We computed two rank-based metrics:

(i) Mean Reciprocal Rank (MRR), which computes the arithmetic mean of reciprocal ranks of all true entities from the ranked list of answers to queries T . and (ii) Hits@ N ($N = 1, 3, 10$), which counts the true entities and calculate their proportion in the truth T in top N sorted predicted answers' list.

$$\text{MRR} = \frac{1}{T} \sum_{i=1}^T \frac{1}{r_i}$$

$$\text{Hits@N} = \frac{1}{T} \sum_{r \in T, r \leq N} \mathbb{I}$$

Datasets We evaluated our method on five widely used benchmark datasets: FB15K (Bordes et al. 2013) is a subset of Freebase, the contents of which are general facts. WN18 (Bordes et al. 2013) is a subset of Wordnet, a database that features lexical relations between words. FB15K-237 (Toutanova and Chen 2015) is a subset of FB15K. WN18RR (Dettmers et al. 2018) is a subset of WN18. YAGO3-10 (Dettmers et al. 2018) mostly describes attributes of persons, and contains entities associated with at least ten different relations.

As was first noted by Toutanova and Chen (2015) that WN18 and FB15k suffer from test leakage through inverse relations: For example, the test set frequently contains triples such as (s, hyponym, o) while the training set contains its inverse (o, hypernym, s). To create a dataset without this property, Toutanova and Chen (2015) introduced FB15K-237, a subset of FB15K where inverse relations are removed. WN18RR was created for the same reason by Dettmers et al. (2018).

The FB15k-237 and WN18RR datasets both include composition relations (e.g. awardnominee/.../nominatedfor), symmetry (e.g. derivationally_related_form in WN18RR), and anti-symmetry (e.g. has_part in WN18RR). The WN18RR dataset includes hierarchical relations such as hypernym and has_part, also_see, similar_to.

Baselines We compared the results on two best performing models 5★E (Nayyeri et al. 2021) and ComplEx (Trouillon et al. 2016).

Our method is implemented in Pytorch¹ and the code² is available online.

Concretely, we restricted the parameters to conjugate of their symmetric counterparts respecting to the diagonal.

5.2 Results

Device We compared all experiments results running on Tesla V100S-PCIE-32GB.

Analysis Since hyper parameters usually affect the performance of KGE models a lot, we did experiments using different hyper parameters settings. σ means standard deviation of experiments' results. Here, we compared σ_{MRR} , which is the standard deviation of MRR to clarify the differences among different models. Furthermore, we adopted **variance homogeneity test** and **two independent-sample t-test** to demonstrate the difference of baseline and our model.

The most important observations are:

- 1) Although we didn't find out the best hyper parameters setting for all baseline models (e.g., 5★E). We observe that in all hyper parameter settings, the proposed conjugated models performs comparable or even little improved than origin model in all big datasets.
- 2) The parameters size and calculation time were reduced are around 0-50% in different baseline models, while the performances are comparable on big datasets without overfitting problem.
- 3) On smaller datasets, reducing parameters sometimes might cause overfitting problem in regularization process, which hurts the performance. (e.g., there is no overfitting problem in ComplEx_{conj} model, but there is in 5★E_{conj} model.)

Moreover, we found other interesting observations:

¹<https://pytorch.org/>

²<https://github.com/FengXincan/dimension>

- 1) ComplEx_{conj} performs comparable with 50% conjugate parameters in all big or small datasets without overfitting problem.
- 2) 5★E Model without regularization (which we call 5★E_{noreg} below) performs very alike with conjugate models 5★E_{conj}.
- 3) The performance of 5★E_{conj} comparing with 5★E is **basically proportional to $\frac{Rel \times Exa}{Ent}$** . The results are comparable in datasets FB15K-237, FB15K, YAGO3-10, but overfitted in dataset WN18RR and WN18.

As was also discussed about by [Ruffinelli et al. 2020](#), when trained appropriately, the relative performance differences between various model architectures often shrinks and sometimes even reverses when compared to prior results. And our conjugate method does not seem to cause that kind of problem except for overfitting occurs. Thus we conclude that:

- 1) **in link prediction, or say it KGE or KGC problem, embedding dimensions are probably conjugated in truth.**
- 2) Using 50% conjugate parameters generally does not hurt model performance or at least directly, while in smaller datasets, sometimes, the parameters reduction in regularization process will cause overfitting problem.

FB15K-237						
Models	MRR	H@1	H@3	H@10	Time	σ_{MRR}
ComplEx						
ComplEx _{conj}						
5★E	0.3349	0.2452	0.3668	0.5162	75s	
5★E _{noreg}	0.3364	0.2465	0.3688	0.5169		
5★E _{conj}	0.3357	0.2457	0.3688	0.5150	27s	
5★E _{dconj}						

Table 1: The results are the average or standard deviation of 5, 2, 2 same experiments, respectively.

WN18RR						
Models	MRR	H@1	H@3	H@10	Time	σ_{MRR}
ComplEx						
ComplEx _{conj}						
5★E	0.4408	0.4137	0.4511	0.4920	35s	
5★E _{noreg}	0.4408	0.4139	0.4501	0.4915		
5★E _{conj}	0.4373	0.4116	0.4455	0.4874	25s	
5★E _{dconj}						

Table 2: The results are the average of 5, 2, 2 same experiments, respectively.

YAGO3-10						
Models	MRR	H@1	H@3	H@10	Time	σ_{MRR}
ComplEx						
ComplEx _{conj}						
5★E	0.5695	0.4964	0.6131	0.7014	555s	
5★E _{noreg}						
5★E _{conj}	0.5705	0.49715	0.6158	0.7008	432s	
5★E _{dconj}						

Table 3: The results are the average of 5, 0, 2 same experiments, respectively.

FB15K						
Models	MRR	H@1	H@3	H@10	Time	σ_{MRR}
ComplEx						
ComplEx _{conj}						
5★E	0.7982	0.7452	0.8341	0.8890	71s	
5★E _{noreg}						
5★E _{conj}	0.7984	0.7457	0.8337	0.8905	48s	
5★E _{dconj}						

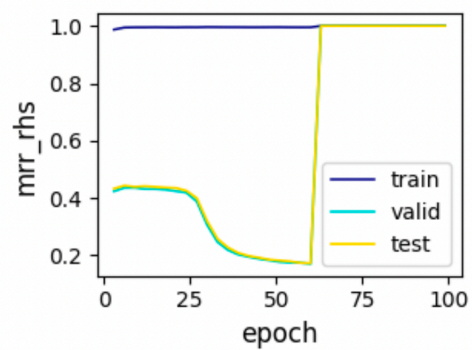
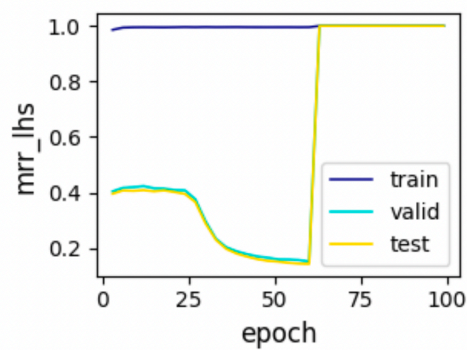
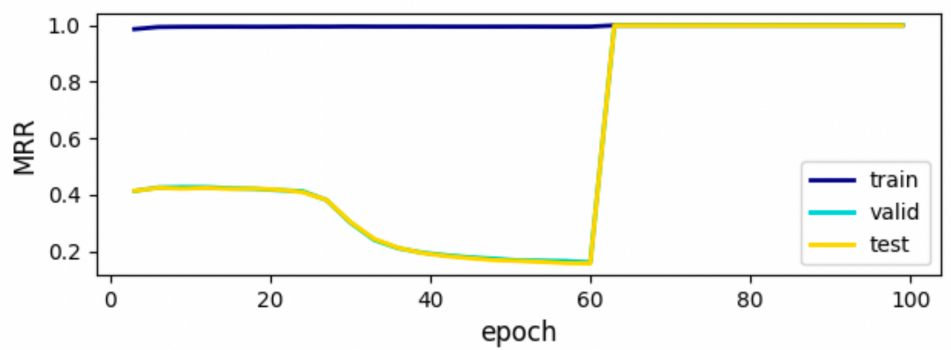
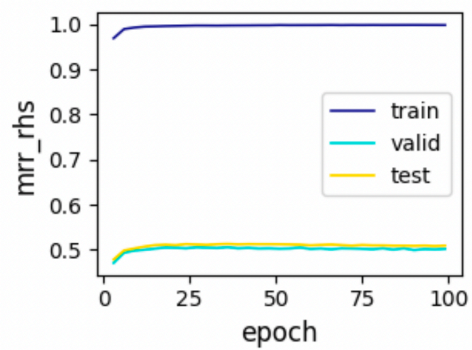
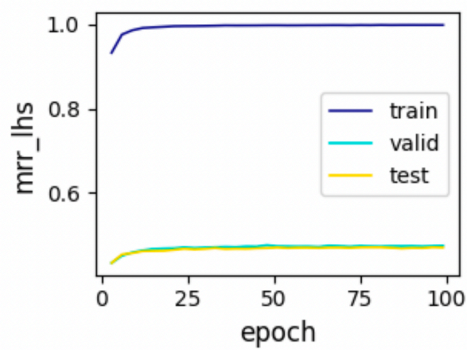
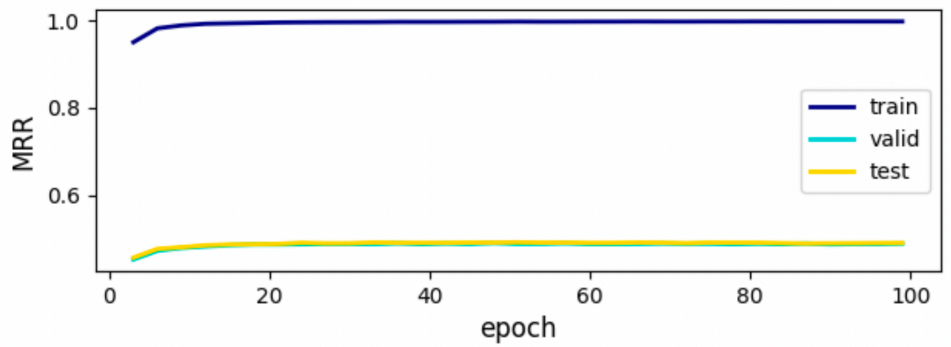
Table 4: The results are the average of 2, 0, 2 same experiments, respectively.

WN18						
Models	MRR	H@1	H@3	H@10	Time	σ_{MRR}
ComplEx						
ComplEx _{conj}						
5★E	0.9442	0.9377	0.9503	0.9535	55s	
5★E _{noreg}						
5★E _{conj}	0.9192	0.8888	0.9497	0.9528	37s	
5★E _{dconj}						

Table 5: The results are the average of 2, 0, 2 same experiments, respectively.

Datasets	Ent	Rel	Exa	$\frac{Rel}{Ent}$	$\frac{Rel \times Exa}{Ent}$
FB15K-237	14541	237	544230	0.01630	8870
WN18RR	40943	11	173670	0.00027	46
YAGO3-10	123188	37	2158080	0.00030	648
FB15K	14951	1345	966284	0.08996	86927
WN18	40943	18	282884	0.00044	124

Table 6: Datasets statistics: Ent: Entities; Rel: Relations; Exa: Examples.



5.2.1 Analysis

5.3 Future Work

The utilize of complex number and geometric projection in knowledge graph embedding models created rather uniform transformations. For better expressiveness of KGE models, we would like to do more research on non-uniform methods to do this task.

6 Appendix

6.1 dimensional relation and geometric relation

Geometric relation can be seen as high dimensional relation. But such high dimensional relation usually make it difficult to analyze its efficiency. So we designed a simple index: dimensional linkage rate = number of combinations / numbers of parameters. Higher dimensional linkage rate indicates higher parameter efficiency. (But this does not necessarily suggests it's a better model.) For example, in the complex model, the dimensional linkage rate = $8/8 = 1$, in our variant model, the dimensional linkage rate = $8/6 \approx 1.33$. Our experiments show that various complex relations might need linked parameters, for example conjugate parameters, to work together. Thus we can leverage this point to reduce calculation in more knowledge graph embedding models.

7 References

- Nayyeri M.; Vahdati S.; Aykul C.; and Lehmann J. 5★ Knowledge Graph Embeddings with Projective Transformations. In *AAAI*. 2021
- Ji S.; Pan S.; Cambria E.; Marttinen P.; and Yu P. S. A survey on knowledge graphs: Representation, acquisition and applications. In *arXiv:2002.00388*. 2020
- Trouillon T.; Welbl J.; Riedel S.; Gaussier É.; and Bouchard G. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*, 2071-2080. 2016
- Chami I.; Wolf A.; Juan D.-C.; Sala F.; Ravi S.; and Ré C. Low-Dimensional Hyperbolic Knowledge Graph Embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 6901-6914. 2020
- Vaswani A.; Shazeer N.; Parmar N.; Uszkoreit J.; Jones L.; Gomez A. N.; Kaiser Ł.; Polosukhin I. Attention Is All You Need. In *arXiv:1706.03762v5*. 2017
- Sun Z.-Q.; Deng Z.-H.; Nie J.-Y.; and Tang J. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*. 2019
- Bordes A.; Usunier N.; Garcia-Duran A.; Weston J.; and Yakhnenko O. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, 2787-2795. 2013
- Ji G.; He S.; Xu L.; Liu K.; and Zhao J. Knowledge graph embedding via dynamic mapping matrix. 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)*, 687-696. 2015

- Lin Y.; Liu Z.; Sun M.; Liu Y.; and Zhu X. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth AAAI conference on artificial intelligence*. 2015
- Nickel M.; Tresp V.; and Kriegel H.-P. A three-way model for collective learning on multi-relational data. In *International Conference on Machine Learning*, pages 809-816. Omnipress. 2011
- Yang B.; Yih W.-t.; He X.; Gao J.; and Deng L. Embedding entities and relations for learning and inference in knowledge bases. In *Conference on Learning Representations (ICLR)*. 2015
- Balazevic I.; Allen C.; and Hospedales T. Multi-relational Poincaré graph embeddings. In *Advances in Neural Information Processing Systems*, 4465-4475. 2019
- Toutanova K.; and Chen D. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, 57-66. 2015
- Dettmers T.; Minervini P.; Stenetorp P.; and Riedel S. Convolutional 2d knowledge graph embeddings. In *Thirty-Second AAAI Conference*. 2018
- Ruffinelli D.; Broscheit S.; and Gemulla R. You can teach an old dog new tricks! On training knowledge graph embeddings. In *The International Conference on Learning Representations*. 2020