Domain and Range Aware Synthetic Negatives Generation for Knowledge Graph Embedding Models

Alberto Bernardi
Accenture Labs, Dublin
alberto.bernardi@accenture.com

Luca Costabello
Accenture Labs, Dublin
luca.costabello@accenture.com

Abstract

Knowledge Graph Embedding models, representing entities and edges in a low-dimensional space, have been extremely successful at solving tasks related to completing and exploring Knowledge Graphs (KGs). One of the key aspects of training most of these models is teaching to discriminate between true statements (*positives*) and false ones (*negatives*). However, the way in which negatives can be defined is not trivial, as facts missing from the KG are not necessarily false and a set of ground truth negatives is hardly ever given. This makes synthetic negative generation a necessity. Different generation strategies can heavily affect the quality of the embeddings, making it a primary aspect to consider. We revamp a strategy that generates corruptions during training respecting the domain and range of relations, we extend its capabilities and we show our methods bring substantial improvement (+10% MRR) for standard benchmark datasets and over +150% MRR for a larger ontology-backed dataset.

1 Introduction

Knowledge Graphs (KGs) are a flexible and scalable data representation paradigm to represent complex relationships between different concepts [1]. Their applicability to model information related to any domain made them reach a significant size, including millions or even billions of edges. This makes manual curation almost impossible and leaves them incomplete and error-prone [2].

In this context, a machine-based approach is necessary to infer missing links and assess the correctness of given links. Pioneer research efforts [3, 4] led to Knowledge Graph Embedding (KGE) models, that learn low-dimensional representations for entities and relations to perform a variety of tasks.

Although some approaches rely uniquely on the true facts that make up a KG, the dominant training approach for KGE models teaches to discriminate between true statements (*positives*) and false ones (*negatives*). However, statements missing in a KG are not necessarily false, and rarely ground truth negatives are provided. This makes the generation of synthetic negatives, or *Negative Sampling* (NS), necessary and determinant during training. A lot of research was carried out, addressing the difficult challenge of coming up with effective and efficient strategies to sample true negatives [5, 6].

Our work extends an existing NS strategy [7] that leverages schema-based information in order to enhance the quality of the negatives. Despite nailing the possibility of using entity type information to generate negatives that are semantically relevant, they neglect how, in case a class has few instances, the likelihood of resampling the same entity from the class increases, so we might end up generating non-diverse and (if the sampled entity coincides with the ground truth one) false negatives. Combining it with random uniform sampling [3], we mitigate this issue while also limiting the risk of generating trivial random negatives. The latter problem, indeed, affects the uniform random sampling applied on its own, especially in KGs where some node types are over-represented.

As an additional contribution, we validate our results on two popular benchmark datasets, where the classes of nodes are inferred based on relation types, and on a biological dataset for which classes are

Bernardi, A. and Costabello, L., Domain and Range Aware Synthetic Negatives Generation for Knowledge Graph Embedding Models (Extended Abstract). Presented at the Third Learning on Graphs Conference (LoG 2024), Virtual Event, November 26–29, 2024.

available as part of the ontology. Importantly, we limit our analysis to four traditional KGE models. Indeed, despite recent works have claimed new state-of-the-art results, the performance gain is often marginal and inflated [8] by computationally expensive training protocols (*e.g.*, 1-N scoring [9], high-dimensional embeddings) that are hardly scalable to real-world applications. Therefore, we re-implement and explore models of manageable size, showing how our approach can lead to significant improvements while competing with the state-of-the-art.

2 Related Work

Knowledge Graph Embedding Models. KGE models learn continuous representations of entities and relations in the KG from the graph topology and its soft regularities (symmetries and asymmetries, concept homophily, long-range relationships...) to solve a wide variety of tasks. We refer the interested reader to [10] for a comprehensive overview.

Negative Sampling. A lot of methods addressed the challenge of generating negative samples from existing KGs. We can broadly categorize them in 4 groups. (i) Static NS methods consider a pool of possible negatives that does not change during training. Within this group we have: the preeminent random sampling, that corrupts either the tail or the head of a triple picking uniformly at random any entity in the KG [3], that might be extended to relations [11] or limited to entities within the mini-batch [12]; probabilistic negative sampling methods, that fix a probability to decide whether to corrupt the head or the tail of a triple [13–16] or to use random negatives or those sampled from a predetermined list [17]; methods that leverage a pre-trained external models (e.g., a pre-trained KGE [18] or Language Model [19]) to generate corruptions that are similar to the entity to corrupt; auxiliary data-based, that leverage type information [7], domain or range information [20, 21], or the semantics [22] of a KG to generate negatives. (ii) Self-Adversarial NS strategies use the KGE embeddings to model a probability distribution from which to sample a set of negatives. For example, [23] makes the distribution of negatives proportional to the score assigned from the KGE model to the negatives, so to guarantee that, as training progresses, the model is influenced more by more plausible negatives. However, this strategy comes with the risk of assigning higher relevance to false negatives. Other works along this line are [24–26]. (iii) Adversarial NS methods leverage an external model, trained in a framework that resembles Generative Adversarial Networks [27], to generate negatives. (iv) Dynamic negative sampling methods, finally, try to inform the generation of negatives by changing their distribution as training progresses to avoid trivial negatives. However, (iii) and (iv) introduce a significant computational overhead. For non-attributive graphs, [28] use a similarity score to sample harder negatives, but its efficient definition in KGs is complicated by their heterogeneity. For more details and negative sampling methods, refer to [5, 6].

3 Methods

3.1 Background and Notation

A Knowledge Graph $\mathcal{G} = \{(s,p,o)\} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ is a set of triples $t = (s,p,o)$, each including a subject (*head*) $s \in \mathcal{E}$, a predicate $p \in \mathcal{R}$, and an object (*tail*) $o \in \mathcal{E}$, where $\mathcal{E}$ and $\mathcal{R}$ are the sets of all entities and relation types, respectively. We refer to the task of predicting unseen triples in a KG as *Link Prediction*. It is formalised in literature as a learning to rank problem, where the objective is learning a scoring function $f : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \to \mathbb{R}$ that, given an input triple $t = (s,p,o)$, assigns a score $f(t) = f((s,p,o)) \in \mathbb{R}$ proportional to the likelihood that the fact t is true. Predictions of true triples $t^+ \in \mathcal{T}^+$ are ranked against predictions for synthetic negatives $t^- \in \mathcal{T}^-$, to gauge how well the model discriminates true from false statements. Finally, we introduce the concept of *class*. If in KGs this concept is formally tied to the definition of a schema or an ontology, we will more broadly refer to it to also indicate the grouping of entities we can infer from the relation types in case the ontology is missing. In these cases, we are going to define two classes for every relation type $p \in \mathcal{R}$: $\texttt{domain}_p \coloneqq \{e \in \mathcal{E} \mid \exists o \in \mathcal{E} \text{ s.t. } (e,p,o) \in \mathcal{G}\}$, $\texttt{range}_p \coloneqq \{e \in \mathcal{E} \mid \exists s \in \mathcal{E} \text{ s.t. } (s,p,e) \in \mathcal{G}\}$.

3.2 Ontology-based Negative Sampling

The dominant approach to NS while training a KGE model is, given a positive triple $t = (s,p,o)$, to corrupt either the subject or the object with η entities sampled uniformly at random from $\mathcal{E}$ [3]. Formally, the set of negatives to sample from is defined as $\mathcal{T}^-_{rand} = \{t^- = (s',p,o) \mid s' \in \mathcal{E}\} \cup$

$\{t^- = (s, p, o') \mid o' \in \mathcal{E}\}$. Notice that $\mathcal{T}^-_{rand} \cap \mathcal{G} \neq \emptyset$, which implies that the above approach might lead to the generation of false negatives. Despite this might slightly slow down the convergence of the training, it allows an extremely efficient generation. The negatives sampled from $\mathcal{T}^-_{rand}$, however, might be heavily affected by an unbalanced class distribution. Indeed, if an entity type is over-represented in a KG, it will also be over-represented in $\mathcal{T}^-_{rand}$. Therefore, entities outside this class respecting the "domain" or "range" of a relation might not appear in any of the sampled corruptions, thus significantly limiting the learning process. To address this issue, we take inspiration from [7], that proposes to corrupt triples respecting the semantics of relations extracted from the ontology (if given) or from the facts in the training data. Formally, given triple $t = (s, p, o)$, the set of negatives for this approach will be denoted $\mathcal{T}^-_{t.c.} = \{(s', p, o) \mid s' \in \texttt{domain}_p\} \cup \{(s, p, o') \mid o' \in \texttt{range}_p\}$, where *t.c.* stands for Type-Constrained.
Unlike [7], we do not limit the NS to $\mathcal{T}^-_{t.c.}$; instead, we also sample a proportion of negatives (determined by a hyper-parameter ν) from $\mathcal{T}^-_{rand}$ (refer to Appendix A for the pseudo-code). Otherwise, in case we had a class including few entities, the corruptions obtained replacing entities within this class would likely be identical and might have led to a significant number of false negatives. In the next sections, we are going to analyse in detail how much this affects KGs with different class distributions and show how our strategy allows to significantly improve results compared to the default approaches.

4 Experiments

4.1 Experimental Settings

Datasets. We evaluate our NS strategy on three KGs: FB15k-237 [29] and WN18RR [9] are two of the most widely used benchmarks for link prediction; Hetionet [30] is a network encoding biological data. Differently from the former two, Hetionet does not come with pre-defined train-test splits. Therefore, we create a test set of 10,000 triples and a validation set of 5,000 extracted at random[1] from the training set so that no entity of the test and validation sets was not present in the training set. A summary of the statistics is available in Appendix A.2.

Evaluation. We evaluate our models in the *filtered* setting: for every target triple $t = (s, p, o)$, we separately corrupt the subject and the object with all the entities in the KG not generating a fact in the training, validation or test sets. We then rank test triples against all corruptions. The metrics used are the usual for link prediction: Mean Reciprocal Rank (MRR) and Hits at N (Hits@N).

Hyperparameter Search. We perform an extensive grid search to validate our results. For FB15k-237 and WN18RR, we explore embedding dimension $k = [200, 350]$, $\eta = [10, 20, 30]$, loss functions [self-adversarial, multi-class NLL]. As optimizer we use Adam [32] with learning rate $lr = [1\text{e}{-2}, 1\text{e}{-3}, 1\text{e}{-4}, 1\text{e}{-5}]$ and $\mathcal{L}^p$-regularization coefficient $\lambda = [1\text{e}{-2}, 1\text{e}{-3}, 1\text{e}{-4}]$. Finally, we test the proportion ν of negatives sampled respecting domain and range constraints: $\nu = [0.0 - 1.0]$ with step 0.1. We run experiments for 1000 epochs on FB15k-237 and 4000 epochs on WN18RR, enabling early stopping monitoring MRR, with patience set to 10 and validation frequency of 25 epochs. For Hetionet we run experiments for 1000 epochs and explore $k = [100, 200]$ and $\eta = [10, 20]$, while we keep the rest of the grid search unchanged.

Baseline. We compare our NS strategy with KGE models trained following the random uniform NS [3] and the Type-Constrained approach [7] we have extended. For a fair comparison, we pick as the best baselines those that share the same (or smaller) embedding size and number of negatives of the best performing model trained with our own NS strategy. The KGE models we consider are TransE [3], DistMult [33], ComplEx-N3 [34, 35] and RotatE [23]. Despite some recent works (among the others [36, 37]) claim new state-of-the-art results, the gain is often marginal and inflated by expensive training protocols. Moreover, the impact of our strategy can be gauged independently on the models considered. We leave for future work the analysis of a wider set of models.

[1] We are aware that properly splitting Hetionet poses multiple challenges [31]. However, the purpose of our analysis is to gauge the relative impact of NS strategies on the performance of the models and not to assess the absolute performance of the models. Therefore, we opted for a simple, random split.

Table 1: Results for the different Negative Sampling (NS) strategies: "Rand." indicates random uniform sampling, "T.C." the Type-Constrained one. All reported results are obtained with AmpliGraph [38]. Random uniform sampling results do not always match SOTA as we deliberately used smaller embedding size and number of negatives for a fair comparison. Best results for each datasets are reported in **bold**, while across models are underlined.

Model	NS	WN18RR				FB15k-237				Hetionet			
		MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
TransE	Rand.	0.25	0.04	0.42	0.56	0.29	0.20	0.33	0.47	0.04	0.02	0.04	0.08
	T.C.	0.19	0.01	0.32	0.47	0.15	0.12	0.17	0.22	0.05	0.03	0.05	0.08
	Ours	0.24	0.04	0.41	0.55	0.31	**0.23**	**0.35**	0.49	0.23	0.15	0.25	0.38
DistMult	Rand.	0.47	0.44	0.48	0.53	0.28	0.18	0.31	0.46	0.08	0.05	0.08	0.14
	T.C.	0.46	0.42	0.49	0.55	0.22	0.16	0.24	0.35	0.22	0.14	0.24	0.36
	Ours	0.46	0.43	0.48	0.53	0.30	0.22	0.33	0.47	0.22	0.14	0.24	0.36
ComplEx-N3	Rand.	0.50	0.46	0.52	0.57	0.29	0.19	0.32	0.48	0.09	0.05	0.16	0.33
	T.C.	0.51	0.47	**0.53**	0.58	0.24	0.17	0.26	0.38	0.24	0.16	0.27	0.40
	Ours	0.51	**0.48**	**0.53**	0.58	**0.32**	**0.23**	**0.35**	0.49	0.25	0.17	0.27	0.41
RotatE	Rand.	0.51	0.47	**0.53**	**0.60**	0.30	0.21	0.33	0.48	0.10	0.06	0.11	0.21
	T.C.	0.51	0.47	**0.53**	**0.60**	0.20	0.13	0.20	0.34	0.11	0.07	0.11	0.17
	Ours	**0.52**	**0.48**	**0.53**	**0.60**	0.31	0.22	**0.35**	**0.50**	**0.26**	**0.18**	**0.29**	**0.42**

4.2 Main Results

The results of our experiments are reported in Table 1. Our NS method outperforms the random approach and the Type-Constrained one in all cases, with the only exceptions of TransE and DistMult on WN18RR. The improvement is substantial in FB15k-237: +10% MRR on the random baseline for ComplEx-N3 and +33% on the type-constrained baseline. Indeed, some of the classes inferred from relation types for FB15k-237 have extremely reduced cardinality (11 classes have just one node, 24 up to 3 nodes, 56 up to 10), thus making the generation of negatives entirely based on type-constraints harmful, as it leads to many false negatives. On the other hand, on Hetionet, our NS strategy leads to impressive gains of well over +150% MRR relative to the random sampling baseline. The definition of classes based on a proper ontology, indeed, allows for the generation of more significant and diverse negatives, not dominated by over-represented classes (*e.g.*, Gene and Biological Process). At the same time, the proportion of random uniform negative samples helps increasing the diversity for classes that have few instances (*e.g.*, Disease and Pharmacological Class), thus motivating the improvement over the T.C. baseline.

As a final remark, the optimal value of η for TransE, DistMult and ComplEx-N3 on WN18RR is 30, while for all the other models is 10. The low (or null) improvement in the performance for the former three models suggests that our strategy impacts more models trained with few negative samples. Indeed, if the number grows, too many domain and range-based negatives can cause issues similar to those outlined for Type-Constrained NS. Random uniform sampling on its own, on the other hand, is enough to yield more meaningful negatives if η grows significantly. This shows how our approach can hugely benefit efficiency, allowing for a reduced number of negative samples. Moreover, the overhead introduced by our strategy is minimal, as ComplEx-N3 run on Hetionet (over 2 million triples) took an average $\sim$5ms per step extra overhead compared to the random uniform sampling. For the detailed list of the optimal hyperparameters refer to Appendix A.4, and for further exploration of the impact of ν on the results refer to Appendix A.5.

5 Conclusion and Future Work

With this work, we have proposed a simple and effective NS strategy, that can significantly enhance the performance of KGE models and help improve the size, and thus the efficiency, of these models. We have validated results on two popular benchmark KGs and on a biological graph that, given an ontology, was affected the most by our method. The results obtained are extremely promising and, given the flexibility of the approach, we leave for further work its application to a broader set of KGE methods. In addition, we plan to extend the experiments to more datasets, to specifically gauge the impact of ontology-derived classes over relation-inferred ones.

References

[1] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia D'amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. Knowledge graphs. 54(4), jul 2021. ISSN 0360-0300. doi: 10.1145/3447772. URL https://doi.org/10.1145/3447772. 1

[2] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, page 601–610, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450329569. doi: 10.1145/2623330.2623623. URL https://doi.org/10.1145/2623330.2623623. 1

[3] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper_files/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf. 1, 2, 3

[4] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. Factorizing YAGO: scalable machine learning for linked data. In Alain Mille, Fabien Gandon, Jacques Misselis, Michael Rabinovich, and Steffen Staab, editors, *Proceedings of the 21st World Wide Web Conference 2012, WWW 2012, Lyon, France, April 16-20, 2012*, pages 271–280. ACM, 2012. doi: 10.1145/2187836.2187874. URL https://doi.org/10.1145/2187836.2187874. 1

[5] Hidetaka Kamigaito and Katsuhiko Hayashi. Comprehensive analysis of negative sampling in knowledge graph representation learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 10661–10675. PMLR, 2022. URL https://proceedings.mlr.press/v162/kamigaito22a.html. 1, 2

[6] Tiroshan Madushanka and Ryutaro Ichise. Negative sampling in knowledge graph representation learning: A review. *CoRR*, abs/2402.19195, 2024. doi: 10.48550/ARXIV.2402.19195. URL https://doi.org/10.48550/arXiv.2402.19195. 1, 2

[7] Denis Krompaß, Stephan Baier, and Volker Tresp. Type-constrained representation learning in knowledge graphs. In Marcelo Arenas, Óscar Corcho, Elena Simperl, Markus Strohmaier, Mathieu d'Aquin, Kavitha Srinivas, Paul Groth, Michel Dumontier, Jeff Heflin, Krishnaprasad Thirunarayan, and Steffen Staab, editors, *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part I*, volume 9366 of *Lecture Notes in Computer Science*, pages 640–655. Springer, 2015. doi: 10.1007/978-3-319-25007-6_37. URL https://doi.org/10.1007/978-3-319-25007-6_37. 1, 2, 3

[8] Daniel Ruffinelli, Samuel Broscheit, and Rainer Gemulla. You CAN teach an old dog new tricks! on training knowledge graph embeddings. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=BkxSmlBFvr. 2

[9] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 1811–1818. AAAI Press, 2018. doi: 10.1609/AAAI.V32I1.11573. URL https://doi.org/10.1609/aaai.v32i1.11573. 2, 3

[10] Jiahang Cao, Jinyuan Fang, Zaiqiao Meng, and Shangsong Liang. Knowledge graph embedding: A survey from the perspective of representation spaces. *ACM Comput. Surv.*, 56(6), mar 2024. ISSN 0360-0300. doi: 10.1145/3643806. URL https://doi.org/10.1145/3643806. 2

[11] Peng Xu and Denilson Barbosa. Investigations on knowledge base embedding for relation prediction and extraction, 2018. 2

[12] Adam Lerer, Ledell Wu, Jiajun Shen, Timothée Lacroix, Luca Wehrstedt, Abhijit Bose, and Alex Peysakhovich. Pytorch-biggraph: A large scale graph embedding system. In Ameet Talwalkar, Virginia Smith, and Matei Zaharia, editors, *Proceedings of the Second Conference on Machine Learning and Systems, SysML 2019, Stanford, CA, USA, March 31 - April 2, 2019*. mlsys.org, 2019. URL https://proceedings.mlsys.org/paper_files/paper/2019/hash/1eb34d662b67a14e3511d0dfd78669be-Abstract.html. 2

[13] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In Carla E. Brodley and Peter Stone, editors, *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada*, pages 1112–1119. AAAI Press, 2014. doi: 10.1609/AAAI.V28I1.8870. URL https://doi.org/10.1609/aaai.v28i1.8870. 2

[14] Qizhe Xie, Xuezhe Ma, Zihang Dai, and Eduard H. Hovy. An interpretable knowledge transfer model for knowledge base completion. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 950–962. Association for Computational Linguistics, 2017. doi: 10.18653/V1/P17-1088. URL https://doi.org/10.18653/v1/P17-1088.

[15] Yi Zhang, Wanhua Cao, and Juntao Liu. A novel negative sample generating method for knowledge graph embedding. In Yunhao Liu, Guoliang Xing, Yuan He, and Gian Pietro Picco, editors, *Proceedings of the 2019 International Conference on Embedded Wireless Systems and Networks, EWSN 2019, Beijing, China, February 25-27, 2019*, pages 401–406. ACM, 2019. URL https://dl.acm.org/citation.cfm?id=3324417.

[16] Wanhua Cao, Yi Zhang, Juntao Liu, and Ziyun Rao. A novel negative sampling based on frequency of relational association entities for knowledge graph embedding. *J. Web Eng.*, 20(6), 2021. doi: 10.13052/JWE1540-9589.2068. URL https://doi.org/10.13052/jwe1540-9589.2068. 2

[17] Vibhor Kanojia, Hideyuki Maeda, Riku Togashi, and Sumio Fujita. Enhancing knowledge graph embedding with probabilistic negative sampling. In Rick Barrett, Rick Cummings, Eugene Agichtein, and Evgeniy Gabrilovich, editors, *Proceedings of the 26th International Conference on World Wide Web Companion, Perth, Australia, April 3-7, 2017*, pages 801–802. ACM, 2017. doi: 10.1145/3041021.3054238. URL https://doi.org/10.1145/3041021.3054238. 2

[18] Bhushan Kotnis and Vivi Nastase. Analysis of the impact of negative sampling on link prediction in knowledge graphs. *CoRR*, abs/1708.06816, 2017. URL http://arxiv.org/abs/1708.06816. 2

[19] Mirza Mohtashim Alam, Md. Rashad Al Hasan Rony, Semab Ali, Jens Lehmann, and Sahar Vahdati. Language model-driven negative sampling. *CoRR*, abs/2203.04703, 2022. doi: 10.48550/ARXIV.2203.04703. URL https://doi.org/10.48550/arXiv.2203.04703. 2

[20] Yuncheng Wang, Xiou Ge, Bin Wang, and C.-C. Jay Kuo. Kgboost: A classification-based knowledge base completion method with negative sampling. *Pattern Recognit. Lett.*, 157: 104–111, 2022. doi: 10.1016/J.PATREC.2022.04.001. URL https://doi.org/10.1016/j.patrec.2022.04.001. 2

[21] Juan Li, Wen Zhang, and Huajun Chen. Incorporating domain and range of relations for knowledge graph completion. In Xiaoyan Zhu, Bing Qin, Xiaodan Zhu, Ming Liu, and Longhua Qian, editors, *Knowledge Graph and Semantic Computing: Knowledge Computing and Language Understanding - 4th China Conference, CCKS 2019, Hangzhou, China, August 24-27, 2019, Revised Selected Papers*, volume 1134 of *Communications in Computer and Information Science*, pages 50–61. Springer, 2019. doi: 10.1007/978-981-15-1956-7_5. URL https://doi.org/10.1007/978-981-15-1956-7_5. 2

[22] Michael Weyns, Pieter Bonte, Bram Steenwinckel, Filip De Turck, and Femke Ongenae. Conditional constraints for knowledge graph embeddings. In Mehwish Alam, Davide Buscaldi, Michael Cochez, Francesco Osborne, Diego Reforgiato Recupero, and Harald Sack, editors, *Proceedings of the Workshop on Deep Learning for Knowledge Graphs (DL4KG2020) co-located with the 17th Extended Semantic Web Conference 2020 (ESWC 2020), Heraklion, Greece, June 02, 2020 - moved online*, volume 2635 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2020. URL https://ceur-ws.org/Vol-2635/paper3.pdf. 2

[23] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=HkgEQnRqYQ. 2, 3

[24] Yingchun Shan, Chenyang Bu, Xiaojian Liu, Shengwei Ji, and Lei Li. Confidence-aware negative sampling method for noisy knowledge graph embedding. In Xindong Wu, Yew-Soon Ong, Charu C. Aggarwal, and Huanhuan Chen, editors, *2018 IEEE International Conference on Big Knowledge, ICBK 2018, Singapore, November 17-18, 2018*, pages 33–40. IEEE Computer Society, 2018. doi: 10.1109/ICBK.2018.00013. URL https://doi.org/10.1109/ICBK.2018.00013. 2

[25] Yongqi Zhang, Quanming Yao, Yingxia Shao, and Lei Chen. Nscaching: Simple and efficient negative sampling for knowledge graph embedding. In *35th IEEE International Conference on Data Engineering, ICDE 2019, Macao, China, April 8-11, 2019*, pages 614–625. IEEE, 2019. doi: 10.1109/ICDE.2019.00061. URL https://doi.org/10.1109/ICDE.2019.00061.

[26] Jingpei Lei, Dantong Ouyang, and Ying Liu. Adversarial knowledge representation learning without external model. *IEEE Access*, 7:3512–3524, 2019. doi: 10.1109/ACCESS.2018.2889481. URL https://doi.org/10.1109/ACCESS.2018.2889481. 2

[27] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2672–2680, 2014. URL https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html. 2

[28] Yuchen Yan, Baoyu Jing, Lihui Liu, Ruijie Wang, Jinning Li, Tarek Abdelzaher, and Hanghang Tong. Reconciling competing sampling strategies of network embedding. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 6844–6861. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/15dc2344ea9bdc01ffb8bb2d692e4018-Paper-Conference.pdf. 2

[29] Kristina Toutanova and Danqi Chen. Observed versus latent features for knowledge base and text inference. In Alexandre Allauzen, Edward Grefenstette, Karl Moritz Hermann, Hugo Larochelle, and Scott Wen-tau Yih, editors, *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality, CVSC 2015, Beijing, China, July 26-31, 2015*, pages 57–66. Association for Computational Linguistics, 2015. doi: 10.18653/V1/W15-4007. URL https://doi.org/10.18653/v1/W15-4007. 3

[30] Daniel Scott Himmelstein, Antoine Lizee, Christine Hessler, Leo Brueggeman, Sabrina L Chen, Dexter Hadley, Ari Green, Pouya Khankhanian, and Sergio E Baranzini. Systematic integration of biomedical knowledge prioritizes drugs for repurposing. *eLife*, 6:e26726, sep 2017. ISSN 2050-084X. doi: 10.7554/eLife.26726. URL https://doi.org/10.7554/eLife.26726. 3

[31] Stephen Bonner, Ian P. Barrett, Cheng Ye, Rowan Swiers, Ola Engkvist, and William L. Hamilton. Understanding the performance of knowledge graph embeddings in drug discovery. *CoRR*, abs/2105.10488, 2021. URL https://arxiv.org/abs/2105.10488. 3

[32] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *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/1412.6980. 3

[33] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In Yoshua Bengio and Yann LeCun, editors, *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. 3

[34] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *ICML*, pages 2071–2080, 2016. 3

[35] Timothee Lacroix, Nicolas Usunier, and Guillaume Obozinski. Canonical tensor decomposition for knowledge base completion. In *International Conference on Machine Learning*, pages 2869–2878, 2018. 3

[36] Ivana Balažević, Carl Allen, and Timothy M Hospedales. Tucker: Tensor factorization for knowledge graph completion. In *Empirical Methods in Natural Language Processing*, 2019. 3

[37] Zhaocheng Zhu, Xinyu Yuan, Mikhail Galkin, Sophie Xhonneux, Ming Zhang, Maxime Gazeau, and Jian Tang. A*net: A scalable path-based reasoning approach for knowledge graphs. *arXiv preprint arXiv:2206.04798*, 2022. 3

[38] Luca Costabello, Alberto Bernardi, Adrianna Janik, Aldan Creo, Sumit Pai, Chan Le Van, Rory McGrath, Nicholas McCarthy, and Pedro Tabacof. AmpliGraph: a Library for Representation Learning on Knowledge Graphs, March 2019. URL https://doi.org/10.5281/zenodo.2595043. 4, 9, 10

A Appendix

A.1 Ontology-based Negative Sampling Pseudo-Code

We provide the pseudo-code of our Negative Sampling strategy in Algorithm 1.

Algorithm 1 Ontology-based Negative Sampling

```
Require: $t = (s, p, o)$, $\mathcal{T}^{-}_{rand}$, $\mathcal{T}^{-}_{t.c.}$, $\nu \in [0, 1]$, $\eta \in \mathbb{N}$.
Ensure: $\mathcal{S}$ set of negatives for $t$ such that $|\mathcal{S}| = \eta$.
    $\eta_{rand} = \lfloor \eta \cdot \nu \rfloor$                ▷ Number of negatives to sample uniformly at random
    $\eta_{t.c.} = \lceil \eta \cdot (1 - \nu) \rceil$            ▷ Number of Ontology-based negatives to sample
    $\mathcal{S}_{rand} \leftarrow$ sample $\left(\mathcal{T}^{-}_{rand}, \eta_{rand}\right)$
    $\mathcal{S}_{t.c.} \leftarrow$ sample $\left(\mathcal{T}^{-}_{t.c.}, \eta_{rand}\right)$
    return $\mathcal{S} \leftarrow \mathcal{S}_{rand} \cup \mathcal{S}_{t.c.}$
```

A.2 Dataset Statistics

Statistics of the three datasets are reported in Table 2.

Table 2: Statistics of the datasets.

Dataset	#Entities	#Relations	#Train	#Valid	#Test	Ontology	#Classes
FB15k-237	15 541	237	272 115	17 535	20 466	✗	474
WN18RR	40 943	18	86 835	3034	3134	✗	36
Hetionet	45 158	24	2 235 197	5000	10 000	✓	11

A.3 Implementation Details

All experiments are implemented with AmpliGraph library [38], using Tensorflow 2.9.0 and Python 3.8.10 in the backend. All our code was run under Ubuntu 20.04 on an Intel Xeon Gold 6226R, 256GB, equipped with Tesla A100 40GB GPUs.

A.4 Optimal Hyperparameters

In Tables 3-4-5 we report the best hyperparameters obtained grid searching our models. k specifies the dimension of the KGE embeddings; η the number of negatives generated for each training triple; lr is the learning rate of the Adam optimizer used in our experiments; ν specifies the proportion of the η negatives that are sampled respecting the domain and range constraints.

A.5 Impact of ν on the Results

As an additional experiment, we report in Figure 1 the performance of the best models for the different scoring functions, modifying the value of ν, the hyper-parameter specifying the proportion of ontology-based negatives to sample for each training triple.

At a first glance, we can notice how, for the same dataset, the impact of ontology-based sampling has a quite consistent behaviour across all the four models, hinting at its applicability to any other scoring function. Moreover, it looks like there is consistency in the behaviour of the same model across neighbouring values of ν. FB15k-237, for example, favours low values of ν, while Hetionet and WN18RR favour higher values. Importantly, this shows how ν is an hyper-parameter easy to tune. Indeed, it will be enough to test a couple of extreme values and a central one to have a good estimate of what the impact of the strategy could be.

Another important observation is the difference even a very small ν can make on the random baseline when there are entity types that are over-represented. It is the case of Hetionet. For this dataset, indeed, we can notice how, even the slightest increase of ν makes the the performance spike. Indeed, as already mentioned in the main text in the paper, ontology-based negatives allows the generation of negatives more pertinent for each triple and not dominated by entities of the over-represented

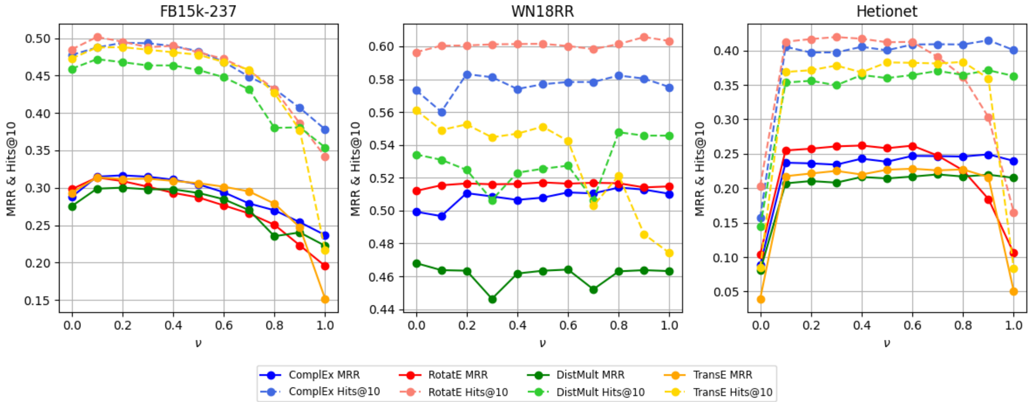

Figure 1: Performance of the different models varying the proportion of ontology-based negatives ν.

types. In FB15k-237 and WN18RR the improvement is less marked as there is no class significantly over-represented.

On the other side of the spectrum of values for ν, we can observe that some models see a significant decrease of the performance as we approach the Type-Constrained baseline ($\nu = 1$). This is definitely dataset-dependent, as it is really evident in FB15k-237 and Hetionet where the classes with very few instances are a significant share of the total. However, it looks like the different scoring function impacts this behaviour. Indeed, TransE and RotatE seems to be more badly affected by an higher number of negatives compared to DistMult and ComplEx. As a mere speculation, we conjecture that this might be related to the different nature of the scoring function: translational/rotational in the former cases, and projective in the latter. We propose to better inspect this behaviour in future work, extending the datasets considered in the analysis and the set of models.

As a final remark, we highlight the inconsistent behaviour of TransE on WN18RR with respect to the other models. The decrease of the Hits@10 observed for higher values of ν is due to the poor expressive power of the model, that thus struggles at making sense of the harder ontology-based negatives. The poor MRR of the model on this dataset is a further element in support (notice we have not reported the MRR of TransE on WN18RR in Figure 1 as it was in the range of $[0.03, 0.04]$ for all νs, including the baseline. Being way below all the other values, it would have compromised the clarity of the plot. However, you can refer to the values reported in Table 1).

A.6 Code Availability

We made the code to reproduce results available at branch `paper/LoG-24-OntologyNegatives` of AmpliGraph [38]. You can access it at `https://github.com/Accenture/AmpliGraph/tree/paper/LoG-24-OntologyNegatives`.

Table 3: WN18RR - Best Model Hyperparameters.

Model	**NS**	**WN18RR**							
		k	η	Loss	lr	Batch Size	$\mathcal{L}^p$-reg (p, λ)	ν	Epochs
TransE	Rand.	350	30	Multi-class NLL	$1\text{e}{-4}$	1000	$p = 2, \lambda = 1\text{e}{-3}$	0.0	1625
	T.C.	350	30	Multi-class NLL	$1\text{e}{-4}$	1000	$p = 2, \lambda = 1\text{e}{-3}$	1.0	1625
	Ours	350	30	Multi-class NLL	$1\text{e}{-4}$	1000	$p = 2, \lambda = 1\text{e}{-3}$	0.2	1500
DistMult	Rand.	350	30	Multi-class NLL	$1\text{e}{-4}$	5000	$p = 2, \lambda = 1\text{e}{-3}$	0.0	950
	T.C.	350	30	Multi-class NLL	$1\text{e}{-4}$	5000	$p = 2, \lambda = 1\text{e}{-3}$	1.0	525
	Ours	350	30	Multi-class NLL	$1\text{e}{-4}$	5000	$p = 2, \lambda = 1\text{e}{-3}$	0.6	650
ComplEx-N3	Rand.	200	20	Multi-class NLL	$1\text{e}{-4}$	5000	$p = 3, \lambda = 1\text{e}{-2}$	0.0	600
	T.C.	200	20	Multi-class NLL	$1\text{e}{-4}$	1000	$p = 3, \lambda = 1\text{e}{-2}$	1.0	975
	Ours	200	20	Multi-class NLL	$1\text{e}{-4}$	1000	$p = 3, \lambda = 1\text{e}{-2}$	0.8	1500
RotatE	Rand.	350	10	Self-Adversarial	$1\text{e}{-4}$	100 000	$p = 3, \lambda = 1\text{e}{-5}$	0.0	1750
	T.C.	350	10	Self-Adversarial	$1\text{e}{-4}$	50 000	$p = 3, \lambda = 1\text{e}{-5}$	1.0	1225
	Ours	350	10	Self-Adversarial	$1\text{e}{-4}$	100 000	$p = 3, \lambda = 1\text{e}{-5}$	0.7	1350

Table 4: FB15k-237 - Best Model Hyperparameters.

Model	**NS**	**FB15k-237**							
		k	η	Loss	lr	Batch Size	$\mathcal{L}^p$-reg (p, λ)	ν	Epochs
TransE	Rand.	200	10	Multi-class NLL	$1\text{e}{-3}$	100 000	$p = 2, \lambda = 1\text{e}{-5}$	0.0	250
	T.C.	200	10	Multi-class NLL	$1\text{e}{-3}$	100 000	$p = 2, \lambda = 1\text{e}{-5}$	1.0	200
	Ours	200	10	Multi-class NLL	$1\text{e}{-3}$	100 000	$p = 2, \lambda = 1\text{e}{-5}$	0.1	1000
DistMult	Rand.	350	10	Multi-class NLL	$1\text{e}{-3}$	100 000	$p = 3, \lambda = 1\text{e}{-3}$	0.0	475
	T.C.	350	10	Multi-class NLL	$1\text{e}{-4}$	100 000	$p = 3, \lambda = 1\text{e}{-3}$	1.0	625
	Ours	350	10	Multi-class NLL	$1\text{e}{-3}$	100 000	$p = 3, \lambda = 1\text{e}{-3}$	0.2	150
ComplEx-N3	Rand.	350	10	Multi-class NLL	$1\text{e}{-3}$	100 000	$p = 3, \lambda = 1\text{e}{-5}$	0.0	275
	T.C.	350	10	Multi-class NLL	$1\text{e}{-4}$	100 000	$p = 3, \lambda = 1\text{e}{-5}$	1.0	550
	Ours	350	10	Multi-class NLL	$1\text{e}{-4}$	100 000	$p = 3, \lambda = 1\text{e}{-3}$	0.2	1000
RotatE	Rand.	350	10	Self-Adversarial	$1\text{e}{-4}$	100 000	$p = 3, \lambda = 1\text{e}{-3}$	0.0	700
	T.C.	350	10	Self-Adversarial	$1\text{e}{-4}$	100 000	$p = 3, \lambda = 1\text{e}{-3}$	1.0	975
	Ours	350	10	Self-Adversarial	$1\text{e}{-4}$	100 000	$p = 3, \lambda = 1\text{e}{-3}$	0.1	500

Table 5: Hetionet - Best Model Hyperparameters.

Model	**NS**	**Hetionet**							
		k	η	Loss	lr	Batch Size	$\mathcal{L}^p$-reg (p, λ)	ν	Epochs
TransE	Rand.	200	10	Multi-class NLL	$1\text{e}{-2}$	100 000	$p = 2, \lambda = 1\text{e}{-5}$	0.0	975
	T.C.	200	10	Multi-class NLL	$1\text{e}{-4}$	100 000	$p = 2, \lambda = 1\text{e}{-3}$	1.0	100
	Ours	200	10	Multi-class NLL	$1\text{e}{-4}$	100 000	$p = 2, \lambda = 1\text{e}{-5}$	0.8	850
DistMult	Rand.	200	10	Multi-class NLL	$1\text{e}{-4}$	100 000	$p = 3, \lambda = 1\text{e}{-5}$	0.0	50
	T.C.	200	10	Multi-class NLL	$1\text{e}{-4}$	100 000	$p = 3, \lambda = 1\text{e}{-3}$	1.0	875
	Ours	200	10	Multi-class NLL	$1\text{e}{-4}$	100 000	$p = 3, \lambda = 1\text{e}{-5}$	0.8	925
ComplEx-N3	Rand.	200	10	Multi-class NLL	$1\text{e}{-4}$	100 000	$p = 3, \lambda = 1\text{e}{-5}$	0.0	25
	T.C.	200	10	Multi-class NLL	$1\text{e}{-4}$	100 000	$p = 3, \lambda = 1\text{e}{-3}$	1.0	750
	Ours	200	10	Multi-class NLL	$1\text{e}{-4}$	100 000	$p = 3, \lambda = 1\text{e}{-3}$	0.8	925
RotatE	Rand.	200	10	Multi-class NLL	$1\text{e}{-3}$	100 000	$p = 3, \lambda = 1\text{e}{-5}$	0.0	25
	T.C.	200	10	Multi-class NLL	$1\text{e}{-4}$	100 000	$p = 3, \lambda = 1\text{e}{-3}$	1.0	375
	Ours	200	10	Multi-class NLL	$1\text{e}{-4}$	100 000	$p = 3, \lambda = 1\text{e}{-5}$	0.3	650