KGrEaT: A Framework to Evaluate Knowledge Graphs via Downstream Tasks

Nicolas Heist*
Sven Hertling*
Heiko Paulheim
nico@informatik.uni-mannheim.de
sven@informatik.uni-mannheim.de
heiko@informatik.uni-mannheim.de
Data and Web Science Group, University of Mannheim
Mannheim, Germany

ABSTRACT

In recent years, countless research papers have addressed the topics of knowledge graph creation, extension, or completion in order to create knowledge graphs that are larger, more correct, or more diverse. This research is typically motivated by the argumentation that using such enhanced knowledge graphs to solve downstream tasks will improve performance. Nonetheless, this is hardly ever evaluated. Instead, the predominant evaluation metrics - aiming at correctness and completeness - are undoubtedly valuable but fail to capture the complete picture, i.e., how useful the created or enhanced knowledge graph actually is. Further, the accessibility of such a knowledge graph is rarely considered (e.g., whether it contains expressive labels, descriptions, and sufficient context information to link textual mentions to the entities of the knowledge graph). To better judge how well knowledge graphs perform on actual tasks, we present KGrEaT – a framework to estimate the quality of knowledge graphs via actual downstream tasks like classification, clustering, or recommendation. Instead of comparing different methods of processing knowledge graphs with respect to a single task, the purpose of KGrEaT is to compare various knowledge graphs as such by evaluating them on a fixed task setup. The framework takes a knowledge graph as input, automatically maps it to the datasets to be evaluated on, and computes performance metrics for the defined tasks. It is built in a modular way to be easily extendable with additional tasks and datasets.

CCS CONCEPTS

• **Computing methodologies** → **Knowledge representation and reasoning**; • **Information systems** → *Semantic web description languages*; **Recommender systems**; **Clustering and classification**.

KEYWORDS

Knowledge Graph, Evaluation Framework, Entity Mapping, Data Mining, Semantic Recommendation

ACM Reference Format:
Nicolas Heist, Sven Hertling, and Heiko Paulheim. 2023. KGrEaT: A Framework to Evaluate Knowledge Graphs via Downstream Tasks. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3583780.3615241

1 INTRODUCTION

1.1 Motivation

Knowledge graphs (KGs) have emerged as a powerful tool for organizing and representing structured knowledge in a machine-readable format. Starting with Google's announcement of the Google Knowledge Graph in 2012[1], research articles have extensively explored the creation [3, 18, 24], extension [10, 14], refinement [22], and completion [1] of KGs, with the aim of producing larger, more accurate, and more diverse graphs. These efforts are driven by the belief that leveraging enhanced KGs can lead to improved performance in downstream tasks. However, comparative evaluations of different KGs w.r.t. their utility for such tasks are rarely conducted.

In the literature, the vast majority of studies concerned with the evaluation of KGs have focused on intrinsic metrics that are working exclusively with the triples of a graph. Several works introduce quality metrics like accuracy, consistency, or trustworthiness and propose ways to determine them quantitatively [4, 8, 16, 33, 35]. Färber et al. [7] and Heist et al. [11] compare KGs with respect to size, complexity, coverage, and overlap. Additionally, they provide guidelines on which KG to select for a given problem.

Another line of work computes extrinsic task-based metrics to evaluate KG embedding approaches. They use a fixed input KG with a fixed evaluation setup while varying only the embedding approach. Frameworks like GEval [23] or kgbench [5] use data mining tasks like classification or regression for the evaluation, others, like Ali et al. [2] evaluate primarily on link prediction tasks.

1.2 Contributions

To address the evaluation gap of extrinsic metrics for KGs, we propose a framework called KGrEaT (**K**nowledge **Gr**aph **E**v**a**luation

*Both authors contributed equally to this research.

[1] https://blog.google/products/search/introducing-knowledge-graph-things-not/

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0124-5/23/10.
https://doi.org/10.1145/3583780.3615241

via Downstream **T**asks).[2] KGrEaT aims to provide a comprehensive assessment of KGs by evaluating them on multiple kinds of tasks like classification, regression, or recommendation. The evaluation results (e.g., the accuracy of a classification model trained with the KG as background knowledge) serve as extrinsic task-based quality metrics for the KG. By defining a fixed evaluation setup in the framework and applying it to multiple KGs, it is possible to isolate the effect of every single KG and compare how useful they are for solving different kinds of tasks. KGrEaT is built in a modular way to be open for extensions from the community like additional tasks or datasets.

Overall, the contributions of this paper are as follows:

- With KGrEaT, we present a framework to judge the utility of KGs using extrinsic task-based metrics (Section 2).
- In our experiments, we demonstrate the capabilities of the framework in an evaluation and comparison of several well-known cross-domain KGs (Section 3).

2 FRAMEWORK

2.1 Purpose and Limitations

KGrEaT is a framework built to evaluate the performance impact of KGs on multiple downstream tasks. To that end, the framework implements various algorithms to solve tasks like classification, regression, or recommendation of entities. The impact of a given KG is measured by using its information as background knowledge for solving the tasks. To compare the performance of different KGs on downstream tasks, we use a fixed experimental setup with the KG as the only variable. The performance indicators may be used to make an informed decision when picking a KG for a given task. Further, they can be used to compare the performance of different versions of a single KG (e.g., during construction or during its life cycle).

The implemented algorithms are not necessarily state-of-the-art because the primary objective is not to measure how well a task can be solved with a given KG in absolute numbers, but rather in comparison to other KGs or different versions of the same KG. Hence, the absolute numbers of the results only have limited expressive power. However, the framework tries to reduce the bias in the results by averaging over multiple preprocessing methods, datasets, and algorithms.

KGrEaT maps the entities of the KG automatically to the entities of the dataset using a set of configurable mappers. Undoubtedly, the quality of this mapping influences the results generated by the framework. But as the mapping procedure is applied similarly to all evaluated KGs, the mapping quality is mainly influenced by the accessibility of the graph (i.e., whether it provides sufficient context information like labels or descriptions for its entities). To reduce the influence of the mapping strategy on the overall results, the framework provides a way to run experiments with multiple mapping approaches (and possibly average over them).

2.2 Design

The framework is designed in a modular way (c.f. Figure 1), making it easy to add additional preprocessing steps, mappers, or tasks. Every step of a stage is implemented as an isolated docker container[3] with its own environment so that additions can be made without any constraints on the programming language. Another advantage of the container-based architecture is the easy distribution of containers via a container hub, eliminating the need for users to build the framework on their own machines.

The manager is responsible for making necessary preparations (e.g., downloading the input data or gathering entities to be mapped), executing the stages (fetching and running containers of the steps), and visualizing the results (e.g., comparing KG performance on various aggregation levels). The `Preprocessing` and `Mapping` stages can be executed in parallel, and the results are then used to execute the `Task` stage. The whole process can be steered via a command line interface (CLI).

The only input to the evaluation process is the KG in the form of RDF files as well as a configuration. The latter defines how the stages should be run (i.e., which steps to execute in which order). Further, every step can be configured in depth to supply relevant hyper-parameters. For example, one can configure how the KG should be mapped to the datasets (e.g., via matching labels) and define an acceptable similarity value for a match.

In the following, we provide details of the three main stages that are executed when running an evaluation of a KG.

2.3 Preprocessing Stage

The `Preprocessing` stage creates all pre-computable artifacts that are needed in the subsequent `Task` stage (e.g., intermediate representations or statistics of the KG). So far, this stage comprises the computation of embeddings (*TransE* [6], *TransR* [17], *DistMult* [34], *RESCAL* [20], *ComplEx* [32] via the DGL-KE framework [37], and RDF2vec [28] via the jRDF2vec framework [25]). Further, it supports the generation of indices for Approximate Nearest Neighbor (ANN) search (via the hnswlib library [19]).

2.4 Mapping Stage

In the `Mapping` stage, the entities of the KG are automatically mapped to the entities in the datasets. So far, a `Same-As` mapper and a `Label` mapper are implemented. The former uses the same-as links of a KG to map its entities to those of the datasets. A dataset may provide URIs for an entity (e.g., from well-known KGs like DBpedia or Wikidata), but it has to provide at least one label. This label is used by the `Label` mapper to find a corresponding entity in the KG. It uses the RapidFuzz library[4] to estimate the similarity of labels via token-based edit distance. Mappers are composable, i.e., they can be executed in sequence. For example, entities are first mapped via same-as links where available, and the remaining entities are mapped via label similarity.

2.5 Task Stage

In the `Task` stage, the task types are executed for all combinations of datasets and algorithms. Table 1 gives an overview of all possible constellations. In total, KGrEaT contains 26 tasks (i.e., combinations of task types and datasets) that are run with one or more algorithms.

[2] https://github.com/dwslab/kgreat

[3] https://www.docker.com

[4] https://github.com/maxbachmann/RapidFuzz

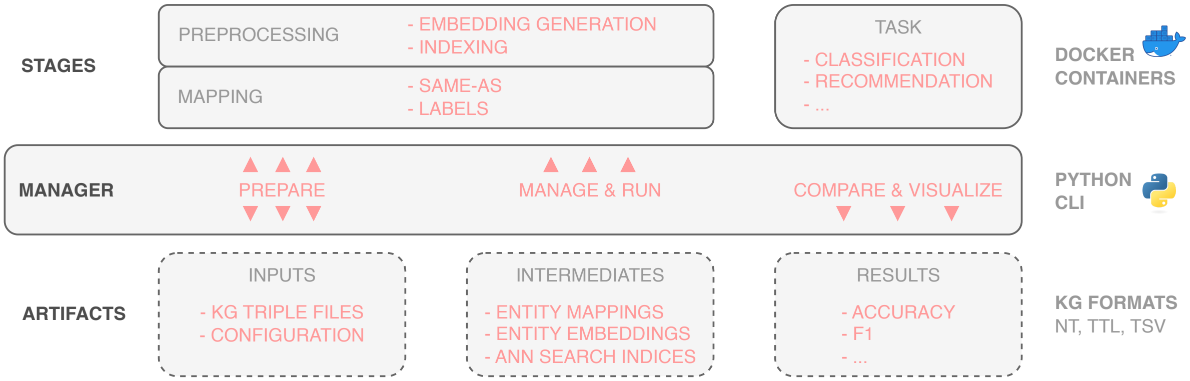

Figure 1: An overview of the KGrEaT framework.

Table 1: Implemented tasks together with their algorithms, datasets, and evaluation metrics.

Task Type	Datasets	Algorithms	Evaluation Metrics
Classification	AAUP, Cities, Forbes, MillionSongDataset MetacriticAlbums, MetacriticMovies, ComicCharacters	Naive Bayes, KNN, SVM	Accuracy ↑
Regression	AAUP, Cities, Forbes, MetacriticAlbums, MetacriticMovies	Linear Regression, KNN, Decision Tree	RMSE ↓
Clustering	Cities2000AndCountries, CitiesAndCountries, Teams, CitiesMoviesAlbumsCompaniesUni, ComicCharacters	DBSCAN, KMeans, Agglomerative Clustering	ARI ↑, NMI ↑, Accuracy ↑
Document Similarity	LP50	Cosine Similarity	Spearman ↑, Pearson ↑
Entity Relatedness	KORE	Cosine Similarity	Kendall's Tau ↑
Semantic Analogies	AllCapitalCountryEntities, CapitalCountryEntities, CityStateEntities, CurrencyEntities	Cosine Similarity	Accuracy ↑
Recommendation	MovieLens, LastFm, LibraryThing	Item-Similarity recommender	F1 ↑

Additionally, the algorithms are executed with multiple hyperparameter settings. How the individual tasks use the KG information is dependent on the task and the implemented algorithm. Generally, the tasks `Classification`, `Regression`, and `Clustering` use embeddings of the KG's entities as features of the models, and the remaining tasks use the distance between the entity embeddings to find related entities.

Several datasets are taken from Ristoski et al. [27] and from the GEval framework [23]. The `Recommendation` datasets MovieLens [9], LastFm[5], and LibraryThing [36] are preprocessed as recommended by Di Noia et al. [21] with the exception of using all entities instead of only those for which a mapping to DBpedia exists. For detailed statistics of all datasets, please refer to the respective publications and the information in the framework.

Every task type comes with suitable evaluation metrics that are computed for every constellation. As some KGs might not contain matches for all entities in the dataset and it would not be fair to compute metrics only over known entities (and discard unknown entities) or only over all entities, the framework reports metrics for both scenarios. Finally, the results can be aggregated over various levels (e.g., over embeddings, algorithms, and datasets) to produce metrics with a reduced bias.

3 EXPERIMENTS

To show the capabilities of KGrEaT, we conduct experiments over multiple large cross-domain KGs and analyze how well they perform on the implemented downstream tasks. We first give an overview of the evaluated KGs, then define the experimental setup, and finally discuss the results.

3.1 Experimental Setup

3.1.1 Knowledge Graphs. We use the following KGs in our experiments:

(1) *DBpedia* [3]: Dumps from 2016-10 and 2022-09[6]
(2) *YAGO* [18]: Version 3
(3) *Wikidata* [24]: Dump from 2023-06-07
(4) *CaLiGraph* [12, 13]: Version 3.1.1
(5) *DBkWik* [15]: Version DBkWik++[7]

[5] http://www.lastfm.com

[6] Using multiple versions allows us to compare not only between different KGs but also between different versions (here: with respect to time) of the same KG.

[7] Combined with DBpedia to also include the well-known entities of Wikipedia

Table 2: Evaluation results of the KGs aggregated by task type and metric. The results of the KGs are given for the dimensions PK (precision-oriented, known entities), PA (precision-oriented, all entities), and RA (recall-oriented, all entities).

Task Type	Metric	DBpedia2016			DBpedia2022			YAGO			Wikidata		CaLiGraph			DbkWik		
		PK	PA	RA	PK	PA	RA	PK	PA	RA	PK	PA	PK	PA	RA	PK	PA	RA
Classification	Accuracy	**0.576**	**0.476**	0.434	0.561	0.433	0.394	0.559	0.464	0.527	0.559	0.366	0.565	0.467	**0.529**	0.539	0.467	0.501
Regression	RMSE	0.693	**1.271**	1.320	0.688	1.287	1.284	0.706	1.331	**0.720**	**0.684**	1.300	0.734	1.321	0.855	0.718	1.287	1.350
Clustering	ARI	0.149	**0.218**	**0.217**	0.115	0.176	0.187	**0.240**	0.216	0.213	0.055	0.076	0.138	0.139	0.125	0.192	0.193	0.199
	NMI	**0.303**	**0.248**	**0.243**	0.272	0.213	0.216	0.278	0.245	0.213	0.178	0.164	0.169	0.190	0.132	0.187	0.200	0.191
	Accuracy	**0.762**	0.614	0.633	0.740	0.556	0.577	0.754	0.560	**0.699**	0.678	0.410	0.708	0.547	0.660	0.691	**0.681**	0.689
Doc. Sim.	Spearman	0.207	0.207	0.207	**0.226**	**0.226**	**0.226**	0.165	0.165	0.160	0.131	0.165	0.203	0.203	0.153	0.200	0.200	0.214
	Pearson	0.294	0.294	0.294	**0.306**	**0.306**	**0.306**	0.235	0.235	0.233	0.241	0.069	0.274	0.274	0.226	0.274	0.274	0.283
	Harm. Mean	0.241	0.241	0.241	**0.257**	**0.257**	**0.257**	0.184	0.184	0.191	0.172	0.103	0.230	0.230	0.180	0.164	0.164	0.241
Ent. Rel.	Kendall's Tau	0.135	0.104	0.109	0.179	0.108	**0.119**	0.012	0.015	0.008	**0.203**	0.071	0.086	0.056	0.078	0.134	**0.119**	0.117
Sem. Analogies	Accuracy	0.253	0.246	**0.261**	**0.265**	**0.249**	0.247	0.221	0.219	0.214	0.001	0.000	0.219	0.187	0.198	0.215	0.212	0.206
Recommend.	F1	0.015	**0.011**	**0.011**	0.014	**0.011**	0.010	0.008	0.006	0.006	**0.021**	0.006	0.013	0.009	0.009	0.011	0.010	0.009

3.1.2 Mapping. We first map the KGs with the `Same-As` mapper where applicable. Then we apply two variants of the `Label` mapper: One with a similarity threshold of 1.0 for high-precision matches and one with a threshold of 0.7 for high recall. For the former, we compute metrics for known entities (**P**recision **K**nown - PK) and for all entities (**P**recision **A**ll - PA); for the latter, being recall-oriented, we report the metrics only for all entities (**R**ecall **A**ll - RA).

3.1.3 Embeddings. To reduce the influence of the different embedding approaches on the overall results, all experiments are executed with four embedding types (*TransE*, *DistMult*, *ComplEx*, and *RDF2vec*). For Wikidata, we could not compute all these embeddings due to the amount of computational resources necessary. Instead, we use pre-computed *TransE* embeddings[8] with a comparable training configuration.

3.1.4 Hardware. All experiments are executed on NVIDIA RTX 2080 Ti graphic cards and Intel Xeon E5 processors (2.6GHz). On average, a full evaluation of a single KG takes roughly 30 hours with 20 hours of embedding computation, 4 hours of mapping, and 6 hours of task execution.

3.2 Results and Discussion

Table 2 shows the final results of our evaluation for the three scenarios *PK*, *PA*, and *RA*. The results are averaged after aggregating over all embeddings, datasets, and algorithms. The complete results of the experiments are publicly available.[9]

For `Classification`, DBpedia2016 shows the best results in the precision setting, while CaLiGraph and YAGO achieve the best results in the recall setting. For `Regression`, both DBpedia versions and Wikidata perform well in the precision setup, while again YAGO and CaLiGraph achieve the best results in the recall setting. The `Clustering` task is solved best by DBpedia2016, YAGO, and DbkWik. For `Document Similarity`, version 2022 of DBpedia is the clear winner. For the `Entity Relatedness` task, using DBpedia2022, Wikidata, or DbkWik as background knowledge produces the best results. `Recommendation` is solved best using DBpedia or Wikidata, `Semantic Analogies` is also solved best by DBpedia.

In general, DBpedia dominates the results to a large extent which may be explained by the fact that some of the datasets used in the framework have been derived from the 2015 version of DBpedia. This might also explain that there is no clear advantage of the 2022 version of DBpedia over the older 2016 version. However, both versions of DBpedia perform strongly on the `Recommendation` task which has no direct relation to DBpedia or even Wikipedia.

Our assumption that the KGs with more entities (YAGO, Wikidata, CaLiGraph, and DBkWik) will have an advantage, especially in the `Recommendation` tasks, did only partially prove to be true. However, they have shown strong performances, especially in recall-oriented settings. A reason for this unsteady performance may lie in the increased complexity of training expressive embeddings for large KGs. In the future, we want to explore this further by running evaluations not only with multiple types of embeddings but also with multiple embedding configurations (e.g., number of trained epochs). Another interesting direction to explore is whether combining two KGs (e.g., by concatenating their entity vectors) yields improved results [31].

4 CONCLUSION AND OUTLOOK

We presented KGrEaT, a framework for evaluating the performance of KGs on multiple downstream tasks. In our experiments, we found that, depending on the task, the performance of the KGs varies enormously. To judge the quality of a KG in its completeness, extrinsic evaluation metrics provided by KGrEaT can serve as a valuable addition to the established intrinsic evaluation criteria.

In the future, we want to improve the framework in various ways, e.g., by providing more embedding methods such as RDF2Vec [29] as well as more tasks like KG Question Answering [30].

Further, we plan to include a more comprehensive mapper that uses all information of an entity (such as comments and relations to other entities). To that end, we transform the entities of the datasets into a small KG which is then mapped to the entities of the KG under evaluation. In such a case, systems participating in the Ontology Alignment Evaluation Initiative (OAEI) [26] may prove useful.

To open the framework for users unfamiliar with programming and docker, we will introduce a graphical user interface, allowing them to analyze KGs in a faster and more intuitive way.

[8] https://torchbiggraph.readthedocs.io/en/latest/pretrained_embeddings.html
[9] https://doi.org/10.5281/zenodo.8050446

REFERENCES

[1] Farahnaz Akrami, Mohammed Samiul Saeef, Qingheng Zhang, Wei Hu, and Chengkai Li. 2020. Realistic re-evaluation of knowledge graph completion methods: An experimental study. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 1995–2010.

[2] Mehdi Ali, Max Berrendorf, Charles Tapley Hoyt, Laurent Vermue, Mikhail Galkin, Sahand Sharifzadeh, Asja Fischer, Volker Tresp, and Jens Lehmann. 2022. Bringing Light Into the Dark: A Large-Scale Evaluation of Knowledge Graph Embedding Models Under a Unified Framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 12 (2022), 8825–8845. https://doi.org/10.1109/TPAMI.2021.3124805

[3] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The Semantic Web: 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007+ ASWC 2007, Busan, Korea, November 11-15, 2007. Proceedings*. Springer, 722–735.

[4] Taiyu Ban, Xiangyu Wang, Lyuzhou Chen, Xingyu Wu, Qiuju Chen, and Huanhuan Chen. 2022. Quality Evaluation of Triples in Knowledge Graph by Incorporating Internal With External Consistency. *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[5] Peter Bloem, Xander Wilcke, Lucas van Berkel, and Victor de Boer. 2021. kgbench: A collection of knowledge graph datasets for evaluating relational and multimodal machine learning. In *The Semantic Web: 18th International Conference, ESWC 2021, Virtual Event, June 6–10, 2021, Proceedings 18*. Springer, 614–630.

[6] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems* 26 (2013).

[7] Michael Färber and Achim Rettinger. 2018. Which knowledge graph is best for me? *arXiv preprint arXiv:1809.11099* (2018).

[8] Junyang Gao, Xian Li, Yifan Ethan Xu, Bunyamin Sisman, Xin Luna Dong, and Jun Yang. 2019. Efficient Knowledge Graph Accuracy Evaluation. *Proc. VLDB Endow.* 12, 11 (2019), 1679–1691. https://doi.org/10.14778/3342263.3342642

[9] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* 5, 4 (2015), 1–19.

[10] Nicolas Heist, Sven Hertling, and Heiko Paulheim. 2018. Language-agnostic relation extraction from abstracts in Wikis. *Information* 9, 4 (2018), 75.

[11] Nicolas Heist, Sven Hertling, Daniel Ringler, and Heiko Paulheim. 2020. Knowledge Graphs on the Web-An Overview. *Knowledge Graphs for eXplainable Artificial Intelligence* (2020), 3–22.

[12] Nicolas Heist and Heiko Paulheim. 2019. Uncovering the Semantics of Wikipedia Categories. In *International Semantic Web Conference*. Springer, 219–236.

[13] Nicolas Heist and Heiko Paulheim. 2021. The CaLiGraph ontology as a challenge for OWL reasoners. *arXiv preprint arXiv:2110.05028* (2021).

[14] Nicolas Heist and Heiko Paulheim. 2023. NASTyLinker: NIL-aware scalable transformer-based entity linker. In *European Semantic Web Conference*. Springer, 174–191.

[15] Sven Hertling and Heiko Paulheim. 2022. DBkWik++- Multi Source Matching of Knowledge Graphs. In *Knowledge Graphs and Semantic Web - 4th Iberoamerican Conference and third Indo-American Conference, KGSWC 2022, Madrid, Spain, November 21-23, 2022, Proceedings (Communications in Computer and Information Science, Vol. 1686)*. Springer, 1–15. https://doi.org/10.1007/978-3-031-21422-6_1

[16] Elwin Huaman, Amar Tauqeer, and Anna Fensel. 2021. Towards knowledge graphs validation through weighted knowledge sources. In *Knowledge Graphs and Semantic Web: Third Iberoamerican Conference and Second Indo-American Conference, KGSWC 2021, Kingsville, Texas, USA, November 22–24, 2021, Proceedings 3*. Springer, 47–60.

[17] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 29.

[18] Farzaneh Mahdisoltani, Joanna Biega, and Fabian Suchanek. 2014. Yago3: A knowledge base from multilingual wikipedias. In *7th biennial conference on innovative data systems research*. CIDR Conference.

[19] Yu A Malkov and Dmitry A Yashunin. 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence* 42, 4 (2018), 824–836.

[20] Maximilian Nickel, Volker Tresp, Hans-Peter Kriegel, et al. 2011. A three-way model for collective learning on multi-relational data.. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, Vol. 11. 3104482–3104584.

[21] Tommaso Di Noia, Vito Claudio Ostuni, Paolo Tomeo, and Eugenio Di Sciascio. 2016. Sprank: Semantic path-based ranking for top-n recommendations using linked open data. *ACM Transactions on Intelligent Systems and Technology (TIST)* 8, 1 (2016), 1–34.

[22] Heiko Paulheim. 2017. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic web* 8, 3 (2017), 489–508.

[23] Maria Angela Pellegrino, Abdulrahman Altabba, Martina Garofalo, Petar Ristoski, and Michael Cochez. 2020. GEval: a modular and extensible evaluation framework for graph embedding techniques. In *The Semantic Web: 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31–June 4, 2020, Proceedings 17*. Springer, 565–582.

[24] Thomas Pellissier Tanon, Denny Vrandečić, Sebastian Schaffert, Thomas Steiner, and Lydia Pintscher. 2016. From freebase to wikidata: The great migration. In *Proceedings of the 25th international conference on world wide web*. 1419–1428.

[25] Jan Portisch, Michael Hladik, and Heiko Paulheim. 2020. RDF2Vec Light–A Lightweight Approach for Knowledge Graph Embeddings. *arXiv preprint arXiv:2009.07659* (2020).

[26] Mina Abd Nikooie Pour, Alsayed Algergawy, Patrice Buche, Leyla Jael Castro, Jiaoyan Chen, Hang Dong, Omaima Fallatah, Daniel Faria, Irini Fundulaki, Sven Hertling, Yuan He, Ian Horrocks, Martin Huschka, Liliana Ibanescu, Ernesto Jiménez-Ruiz, Naouel Karam, Amir Laadhar, Patrick Lambrix, Huanyu Li, Ying Li, Franck Michel, Engy Nasr, Heiko Paulheim, Catia Pesquita, Tzanina Saveta, Pavel Shvaiko, Cássia Trojahn, Chantelle Verhey, Mingfang Wu, Beyza Yaman, Ondrej Zamazal, and Lu Zhou. 2022. Results of the Ontology Alignment Evaluation Initiative 2022. In *Proceedings of the 17th International Workshop on Ontology Matching (OM 2022) (CEUR Workshop Proceedings, Vol. 3324)*. CEUR-WS.org, 84–128. https://ceur-ws.org/Vol-3324/oaei22_paper0.pdf

[27] Petar Ristoski, Gerben Klaas Dirk De Vries, and Heiko Paulheim. 2016. A collection of benchmark datasets for systematic evaluations of machine learning on the semantic web. In *The Semantic Web–ISWC 2016: 15th International Semantic Web Conference, Kobe, Japan, October 17–21, 2016, Proceedings, Part II 15*. Springer, 186–194.

[28] Petar Ristoski and Heiko Paulheim. 2016. Rdf2vec: Rdf graph embeddings for data mining. In *The Semantic Web–ISWC 2016: 15th International Semantic Web Conference, Kobe, Japan, October 17–21, 2016, Proceedings, Part I 15*. Springer, 498–514.

[29] Petar Ristoski and Heiko Paulheim. 2016. Rdf2vec: Rdf graph embeddings for data mining. In *The Semantic Web–ISWC 2016: 15th International Semantic Web Conference, Kobe, Japan, October 17–21, 2016, Proceedings, Part I 15*. Springer, 498–514.

[30] Nadine Steinmetz and Kai-Uwe Sattler. 2021. What is in the KGQA benchmark datasets? Survey on challenges in datasets for question answering on knowledge graphs. *Journal on Data Semantics* 10, 3-4 (2021), 241–265.

[31] Steffen Thoma, Achim Rettinger, and Fabian Both. 2017. Towards holistic concept representations: Embedding relational knowledge, visual attributes, and distributional word semantics. In *The Semantic Web–ISWC 2017: 16th International Semantic Web Conference, Vienna, Austria, October 21–25, 2017, Proceedings, Part I 16*. Springer, 694–710.

[32] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International conference on machine learning*. PMLR, 2071–2080.

[33] Xiangyu Wang, Lyuzhou Chen, Taiyu Ban, Muhammad Usman, Yifeng Guan, Shikang Liu, Tianhao Wu, and Huanhuan Chen. 2021. Knowledge graph quality control: A survey. *Fundamental Research* 1, 5 (2021), 607–626. https://doi.org/10.1016/j.fmre.2021.09.003

[34] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1412.6575

[35] Amrapali Zaveri, Anisa Rula, Andrea Maurino, Ricardo Pietrobon, Jens Lehmann, and Sören Auer. 2012. Quality assessment for linked open data: A survey. *Sem. Web* 1 (2012), 1–5.

[36] Tong Zhao, Julian McAuley, and Irwin King. 2015. Improving latent factor models via personalized feature projection for one class recommendation. In *Proceedings of the 24th ACM international on conference on information and knowledge management*. 821–830.

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