

# Empowering Small-Scale Knowledge Graphs: A Strategy of Leveraging General-Purpose Knowledge Graphs for Enriched Embeddings

Albert Sawczyn, Jakub Binkowski, Piotr Bielak, Tomasz Kajdanowicz

Wrocław University of Science and Technology  
wybrzeże Stanisława Wyspiańskiego 27  
albert.sawczyn@pwr.edu.pl

## Abstract

Knowledge-intensive tasks pose a significant challenge for Machine Learning (ML) techniques. Commonly adopted methods, such as Large Language Models (LLMs), often exhibit limitations when applied to such tasks. Nevertheless, there have been notable endeavours to mitigate these challenges, with a significant emphasis on augmenting LLMs through Knowledge Graphs (KGs). While KGs provide many advantages for representing knowledge, their development costs can deter extensive research and applications. Addressing this limitation, we introduce a framework for enriching embeddings of small-scale *domain-specific* Knowledge Graphs with well-established *general-purpose* KGs. Adopting our method, a modest *domain-specific* KG can benefit from a performance boost in downstream tasks when linked to a substantial *general-purpose* KG. Experimental evaluations demonstrate a notable enhancement, with up to a 44% increase observed in the *Hits@10* metric. This relatively unexplored research direction can catalyze more frequent incorporation of KGs in knowledge-intensive tasks, resulting in more robust, reliable ML implementations, which hallucinates less than prevalent LLM solutions.

**Keywords:** knowledge graph, knowledge graph completion, entity alignment, representation learning, machine learning

## 1. Introduction

In recent years, machine learning has started to be commonly used in numerous applications. Recent advancements in Natural Language Processing, especially the emergence of Large Language Models (Devlin et al., 2019; Brown et al., 2020; Touvron et al., 2023), have sparked the adoption of AI-based solutions in many real-world circumstances. The ever-growing demand for automating knowledge-intensive tasks (Petroni et al., 2021) motivates many researchers to utilize NLP solutions, but these are prone to errors (Adlakha et al., 2023; Bowman, 2023). Knowledge-intensive tasks are characterized by the need for comprehensive understanding and reasoning, meaning they require AI models to go beyond surface-level data and tap into a wealth of interconnected information, much like a human expert would (Lewis et al., 2020), while also accounting for the frequent updates of its knowledge. Examples of knowledge-intensive tasks include question-answering systems, fact-checking, information retrieval, and recommendation systems (Petroni et al., 2019). To deal with the peculiarities of such a setting, one might leverage Knowledge Graphs (KGs) that hold knowledge specific to the domain of interest. That is why there were several attempts to combine Language Models with KGs (Colon-Hernandez et al., 2021). However, building comprehensive KGs is an expensive and complex task (Sequeda and Las-

ila, 2021), thus often avoided, which might hinder the development of many endeavours. Therefore, to foster the further development of KG-based systems, we propose a framework that combines well-developed KGs with smaller and still-developing domain-specific KGs.

Creating a KG is a resource-intensive process that requires a significant investment of time and capital. This makes it particularly challenging for small to medium-sized teams, startups, or groups in academia, who often face resource constraints (Ji et al., 2022). A small, developing KG might be insufficient as such a graph often suffers from limited relational structures, sparse entity interactions, and reduced contextual information. This scarcity of data can limit the model's ability to generalize well, capture intricate semantic relationships, or make accurate predictions. To leverage the full potential of Knowledge Graphs and enable their early adoption, we propose an unexplored approach – to focus on creating a smaller, *domain-specific KG (DKG)*, which addresses specific needs, and combining it with a broader but well-developed *general-purpose KG (GKG)*. This enables small KGs to immediately leverage additional knowledge in downstream tasks, eventually leading to better performance. This concept is visualized in Figure 1, i.e., we automatically link *domain-specific* KG to *general-purpose* KG using *entity alignment* (or *alignment* for short) and *linking* operations, which combine two KGs by matching their nodes and

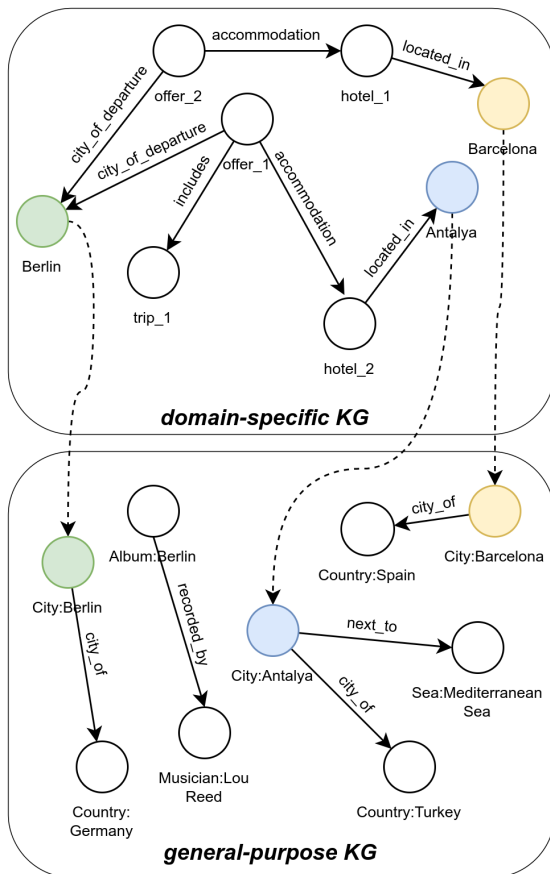


Figure 1: Diagram presenting two aligned and linked Knowledge Graphs: *domain-specific* (upper), *general-purpose* (bottom). Artificial links, marked with dashed lines, connect two KG.

putting artificial links between them. Further, we train a translation-based embedding method in a KG completion task. We demonstrate that training such a model on the linked KGs can enhance performance in a downstream task of KG completion on the *domain-specific* graph compared to a scenario without *general-purpose* KG.

We summarize our contributions as follows:

1. We propose a generic and modular framework for enriching small and *domain-specific* KGs with *general-purpose* KGs using *alignment* and *linking* operation. The framework is tailored to real-world scenarios.
2. We propose an evaluation methodology for experiments and conduct empirical studies on synthetic and real-world scenarios. In both settings, evaluation is conducted under rigorous conditions, i.e., DKG lacks many nodes and links to perform well on a downstream task.
3. We propose a weighted loss function to mitigate entity alignment's negative effects between two considered KGs.

The work adheres to the widely-renowned reproducibility standards. We publish full implementation and data (datasets, models) in the GitHub repository<sup>1</sup>, and experiment tracking logs on Weights&Biases (Biewald, 2020)<sup>2</sup>.

## 2. Related Work

To leverage a Knowledge Graph in a downstream task, like KG completion, one has to obtain representations (Bengio et al., 2013) of its entities (and relations), which are feature vectors describing these objects (Hamilton et al., 2017), similar to (contextual) word embeddings in Natural Language Processing. This pertains to using standalone KG or fusing it with Large Language Model (Colon-Hernandez et al., 2021). Recently, learning representation on graphs, including Knowledge Graphs, gained much interest in the community, primarily due to the development of Graph Neural Networks (Gilmer et al., 2017; Kipf and Welling, 2017). Representation learning of KGs aims to map entities and relationships into low-dimensional vector spaces, thereby indirectly capturing their semantic meaning (Bengio et al., 2013). However, unlike the most prevalent setup of processing homogeneous graphs, learning representations on KGs requires tackling two main challenges. First, the vast size of KGs poses issues as most established methods lose efficacy when handling billions of data points. Second, KGs often lack completeness due to inherent imperfections in their creation process (Pujara et al., 2017).

Nonetheless, several groups of representation learning methods for KGs can be found in the literature. Among these groups, translation-based, GNN-based (Zhang et al., 2022), and factorization-based (Trouillon et al., 2017; Balazevic et al., 2019) methods stand out the most. The TransE model (Bordes et al., 2013) is one of the most common translation-based models in KG representation learning. It operates on the principle of representing relations as translations in the embedding space, i.e., the embedding of the edge should move the embedding of the head entity to end up with an embedding of the tail entity. Due to several limitations of TransE, models such as RotatE (Sun et al., 2019), BoxE (Abboud et al., 2020), and QuatE (Zhang et al., 2019) were proposed. For instance, RotatE leveraged complex space, and the representation of each relation is a rotation from head to tail entity. Thus, RotatE can encapsulate information about symmetry, anti-symmetry, inver-

<sup>1</sup><https://github.com/graphml-lab-pwr/empowering-small-scale-kg>

<sup>2</sup><https://wandb.ai/graph-ml-lab-wust/empowering-small-scale-kg>

sion, and composition in the learned representations (Sun et al., 2019).

The requirements of KGs for learning the attribute features and structural features of entities and relationships can be met by GNN-based methods, which integrate the topology and attribute information. An early yet still prevalent approach is the Relational Graph Convolutional Networks (R-GCN) (Schlichtkrull et al., 2018), a relation-aware variant of GCN. Later significant methods include CompGCN (Vashishth et al., 2019), KB-GAT (Nathani et al., 2019). While GNNs showed promising performance in several studies, we chose the family of translation-based methods for learning representations on combined graphs due to their comparable performance to GNN models with higher stability and shorter learning times (Ali et al., 2022).

Besides, in our work, we propose combining two graphs by entity alignment and linking operation before learning representations of entities. Given two KGs, alignment aims to automatically find matching pairs of entities corresponding to the same real-world entity (Zhang et al., 2022), and linking creates artificial connections between aligned entities. While several approaches for data-driven alignment (Zeng et al., 2021) were proposed, including MTransE (Chen et al., 2017), bootstrapping (Bordes et al., 2013), or AttrE (Trisedya et al., 2019), current benchmarks for assessing these methods exhibit notable restrictions: assumptions of bijection, a lack of name variety, and small scale of graphs. These limitations starkly contrast with the realities encountered in real-world applications, which tend to be larger in scale, more diverse, and far from a 1:1 mapping (Zhang et al., 2022). In our work, we accept that alignment methods might be imperfect and link two graphs based on simple similarity scores between entities. Instead of optimizing loss terms related to alignment, as done in other studies (Liu et al., 2022; Sun et al., 2017), we directly optimize KG completion loss over linked KGs. The previous literature tackled graph linking, but not our approach for embedding enrichment. While aligned with the semantic web vision (Gruber, 1995; Berners-Lee et al., 1998), our focus goes beyond traditional discourse. Hence, to the best of our knowledge, this is the first investigation of this type.

### 3. Methodology

In this work, we propose learning Knowledge Graph completion in a small-scale domain-specific KG by connecting it with a larger general-purpose KG by alignment and linking operation. Further, such linked KG is fed to a downstream task model,

i.e., KG completion. The entire framework is schematically presented in Figure 2, and below, we formally introduce each component.

#### 3.1. Notation

We denote Knowledge Graph as a multi-relational heterogeneous graph  $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$ , consisting of a set of entities  $\mathcal{E}$ , a set of relation predicates  $\mathcal{R}$ , and a set of triples (facts)  $\mathcal{T}$ . A single triple  $(h, r, t) \in \mathcal{T}$  indicates a relation predicate  $r$  between two entities, a head entity  $h$  and a tail entity  $t$ , where  $h, t \in \mathcal{E}$  and  $r \in \mathcal{R}$  (Hamilton et al., 2017).

#### 3.2. Alignment and Linking

First, we define *entity alignment* in the context of two KGs: a domain-specific ( $d$ ) one and a general-purpose ( $g$ ) one:

$$\mathcal{G}_d = (\mathcal{E}_d, \mathcal{R}_d, \mathcal{T}_d), \mathcal{G}_g = (\mathcal{E}_g, \mathcal{R}_g, \mathcal{T}_g)$$

*Alignment* aims to discover entity pairs  $(e_i^{(d)}, e_j^{(g)})$ ,  $e_i^{(d)} \in \mathcal{E}_d, e_j^{(g)} \in \mathcal{E}_g$  where  $e_i^{(d)}$  and  $e_j^{(g)}$  correspond to the same real-world entity. In other words,  $e_i^{(d)}$  and  $e_j^{(g)}$  are aligned entities (Zeng et al., 2021). We propose a simple strategy for alignment, detailed below.

We observe that each entity has assigned a word label, such as "car", and we can vectorize it using a text embedding method of our preference. Further, to achieve the representation  $x(e_i)$  for each entity  $e_i$ , we concatenate three terms: embedding of entity label, mean embedding of outgoing neighbours' labels and mean embedding of ingoing neighbours' labels:

$$x(e_i) = \left[ \phi(e_i) \parallel \sum_{e_u \in \mathcal{O}(e_i)} \frac{\phi(e_u)}{|\mathcal{O}(e_i)|} \parallel \sum_{e_v \in \mathcal{I}(e_i)} \frac{\phi(e_v)}{|\mathcal{I}(e_i)|} \right],$$

where embedding  $\phi(\cdot)$  is a pre-trained text embedding,  $\parallel$  denotes the concatenation operator,  $\mathcal{O}(e_i)$  returns outgoing nodes and  $\mathcal{I}(e_i)$  returns ingoing nodes.

Then, for each entity in the DKG, we find  $k$  nearest entities in the GKG, based on  $x(\cdot)$  embeddings, where  $k$  is a hyperparameter. The nearest neighbour search leverages a vector distance metric, e.g., euclidean, cosine. Further, for any pair of found neighbours  $e_i^{(d)}$  and  $e_j^{(g)}$ , we create an artificial triple  $(e_i^{(d)}, r_l, e_j^{(g)}) \in \mathcal{T}_l$  between two KGs and call it *linking*. The linked KG is formalized as:

$$\mathcal{G}_l = (\mathcal{E}_d \cup \mathcal{E}_g, \mathcal{R}_d \cup \mathcal{R}_g \cup \{r_l\}, \mathcal{T}_d \cup \mathcal{T}_g \cup \mathcal{T}_l),$$

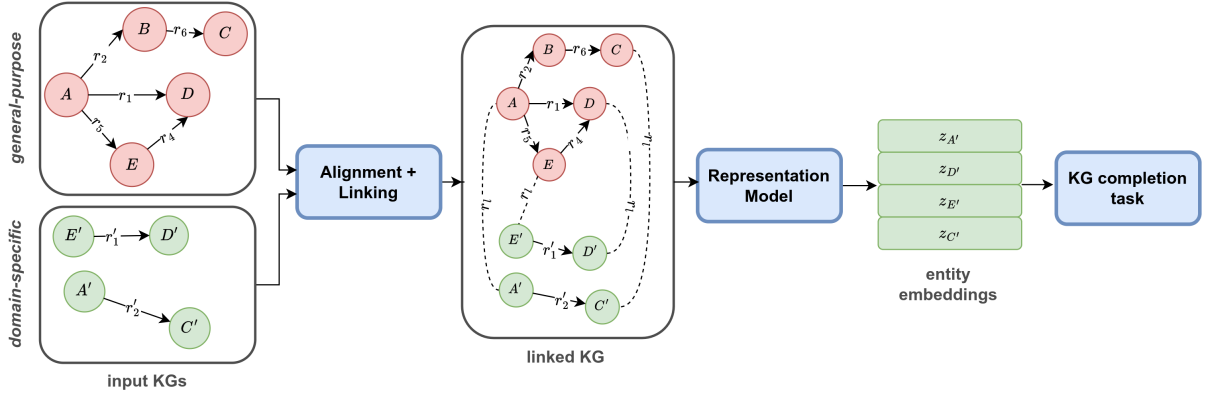


Figure 2: Overview diagram of the proposed framework’s pipeline.

### 3.3. Representation Model

Representation learning aims to learn a model that extracts relevant and semantically rich features from the underlying data. In the context of knowledge graphs, we formalize this task as learning (parametrized) function  $f_\theta : \mathcal{G}_l \rightarrow Z$ , where  $Z \in \mathbf{R}^{N \times d}$  is a matrix of entity representations, where  $N$  is the number of entities, and  $d$  is a dimensionality of the representation. In other words, we expect to optimize  $\theta$  under a particular loss function. It is worth noting that Representation Model can be trained end-to-end with KG completion in a supervised setting (Hamilton et al., 2017), which is also our case. We present the particular choice for the representation model in Section 4.3.

### 3.4. KG completion and Weighted Loss function

In the KG completion task, for every pair of entities in KG, the model predicts the existence of each possible relation between them (Hamilton et al., 2017). During training, the model treats existing relations as positive examples, and to avoid collapse, negative relations are sampled (non-existing ones). In our setting, we assume that the alignment is imperfect. Hence, we introduce a loss function which differently weights relations arising from the linking operation by the distance score used in nearest neighbour search, as shown in Equation (1):

$$\mathcal{L}(\mathcal{T}, \tilde{\mathcal{T}}; \theta) = \sum_{t \in \mathcal{T}_d \cup \mathcal{T}_g} L(t, \tilde{\mathcal{T}}; \theta) + \sum_{s \in \tilde{\mathcal{T}}_l} w_s \cdot L(s, \tilde{\mathcal{T}}; \theta), \quad (1)$$

where  $\mathcal{L}(\cdot)$  denotes loss function for the entire model,  $L(\cdot)$  is a loss function of the Representation Model, and  $\tilde{\mathcal{T}}$  represents the sampled negative relations, and  $w_s = 1/(1 + \text{distance}(x(e_i), x(e_j)))$  represents similarity score between representation of

aligned entities  $e_i$  and  $e_j$ .

## 4. Experiments

We proposed a framework that performs effectively in real situations, mainly when DKG are much smaller than GKG. Since existing datasets do not capture this scenario, we designed a custom evaluation procedure to fit this context. In particular, two distinct scenarios were implemented: synthetic and real-world.

### 4.1. Datasets

In the experiments, we leveraged common KG datasets which emulate the realistic conditions of size diversity, domain overlapping, and mapping possibilities. **WN18RR** (Dettmers et al., 2018) – derived from WordNet (Miller, 1995), a lexical database of English focuses on semantic relations between words. **FB15k-237** (Toutanova and Chen, 2015) – sourced from Freebase, a knowledge graph that spans encyclopedic topics. **WD50K** (Galkin et al., 2020) – an encyclopedic KG extracted from Wikidata (Vrandečić and Krötzsch, 2014) based on the seed nodes corresponding to entities from FB15k-237. **ConceptNet** (Speer et al., 2017) – semantic network encapsulating commonsense knowledge (we used version 5.7.0 and extracted the English segment of this KG). **YAGO3-10** (Mahdisoltani et al., 2015) – a commonsense KG sourced from multiples sources, including WordNet (Miller, 1995).

To access textual labels of entities, we used decoded versions of WN18RR, FB15k-237, and YAGO3-10, which are available in the repository<sup>3</sup>. For WD50K, we utilized Wikidata API<sup>4</sup> to map en-

<sup>3</sup>[https://github.com/villmow/datasets\\_knowledge\\_embedding](https://github.com/villmow/datasets_knowledge_embedding)

<sup>4</sup>[https://www.wikidata.org/wiki/Wikidata:REST\\_API](https://www.wikidata.org/wiki/Wikidata:REST_API)

tity IDs to their textual labels. ConceptNet inherently provides textual labels for its entities, eliminating the need for decoding. Finally, we removed all terms used for identity identification, e.g., `absorb.v.01`  $\rightarrow$  `absorb`. The detailed statistics of the used datasets are presented in Appendix B: Table 3.

## 4.2. Evaluation procedure

An approach we adopted involves taking a dataset and then sampling it for use as the DKG that we used for both synthetic and real-world scenarios we describe below.

### 4.2.1. Sampling

To simulate the early stages of KG development, we introduce three distinct graph sampling strategies:

**Triple sampling** draws triples (facts) at random. While this directly affects the number of facts in the sampled KG, it retains all entities and relations in the chosen triples. However, this might result in some entities or relations being disconnected.

**Node sampling** selects entities and all triples associated with it. This approach often produces a more cohesive subgraph but may exclude distant interactions.

**Relation sampling** selects relations and all triples containing this relation. This method focuses on preserving specific interactions, potentially leading to diverse entities included, but only via the selected relations.

Each strategy is parameterized by the probability  $p$  of keeping a triple, node, or relation in the sampled graph, thus determining the size of the resulting DKG. It offers precise control over the sampling intensity, allowing us to craft KGs of varying sizes and complexities to mimic different stages in developing a KG. The detailed statistics of the sampled datasets are presented in Appendix B: Table 4.

Given the standard transductive setting, we added an initial step to the sampling procedure to ensure that all entities and relations from the validation (val) and testing (test) subsets are preserved in a sampled dataset. This allowed to compare across varying sampling parameters. For triple sampling, this entails having at least one triple for every entity and relation from the val/test. In node sampling, for every entity from these subsets, we ensure the preservation of at least one connected entity, guaranteeing the inclusion of at least one associated triple. However, depending on the value of  $p$ , retaining all val/test data might be impossible, leading to varying minimum  $p$  values across datasets. Notably, in relation sampling, retaining the val/test subsets is impossible. Due to that, performance metrics derived from these

sampled subsets should be interpreted cautiously, avoiding direct comparisons among themselves and with other datasets.

### 4.2.2. Scenarios

**Synthetic scenario** In this scenario, the GKG is the original version of the graph, while the sampled version is taken as DKG. Primarily, the synthetic scenario provides a controlled environment that mimics realistic conditions of non-proportional graphs. It guarantees that the knowledge domains of the two KGs overlap perfectly, ensuring that a direct 1:1 alignment is feasible. This controlled alignment, in turn, provides a more straightforward context to evaluate and understand the properties of our proposed framework.

**Real-world scenario** Here, an external, distinct KG takes on the role of the GKG, i.e., the two KGs are inherently different. This mirrors realistic scenarios where it is needed to enhance the efficacy of an underdeveloped knowledge graph by enriching it with a more extensive and well-established KG. For instance, consider the sampled WN18RR linked to ConceptNet, which encompasses knowledge spanning the domain of WordNet. Unlike the synthetic scenario, this real-world setup does not offer the same level of control for the experimental phase.

## 4.3. Experimental setup

### 4.3.1. Model setup

To perform alignment, we selected fastText word embeddings (Joulin et al., 2016), as these turned out to be effective in the KG domain (Liu et al., 2022). During linking, we performed  $k$  nearest neighbours search based on the Euclidean distance, leveraging the FAISS library (Johnson et al., 2019). Further, as a Representation Model, we utilized RotatE (Sun et al., 2019), which optimizes lookup embeddings for each entity and relation. We trained RotatE end-to-end with the KG completion task (without updating FastText embeddings). We considered two variants of the loss function: the first setting involves the weighted loss function from Eq. (1) setting both  $L(\cdot)$  terms to self-adversarial negative sampling loss (Sun et al., 2019); the second uses standard loss that is equivalent to setting all weights  $w = 1.0$  in the weighted loss. To optimize the model, we leveraged the Adam optimizer (Kingma and Ba, 2014).

Based on the insights from Ali et al., 2022 combined with our empirical observations, the parameters were established as follows: the embedding dimension was set at 256, the batch size at 512,

the learning rate at 0.004, the margin at 9, the adversarial temperature at 0.34, for every positive 33 negatives were generated, and the models were trained for 200 epochs.

During the synthetic scenario, the parameter  $k$  was searched across values  $\{1, 2, 3\}$ . Based on the results of this exploration,  $k$  was set to 1, as it emerged as the most promising value for a real-world scenario. Additionally, due to the large size of GKGs in the real-world scenario, we considered cropping them to include only aligned entities and their  $c$ -hop neighborhood. This approach reduced experiment time and, as our empirical results showed, enhanced performance. Based on preliminary results, we defined specific cropping strategies ( $c = \infty$  meaning not cropping) for each (DKG, GKG) pair, and then searched to select the best-performing one. These strategies are as follows: for (WN18RR, FB15k-237)  $c = \infty$ ; for (FB15k-237, YAGO3-10),  $c \in \{1, 2\}$ ; and for (WD50K, FB15k-237), we set  $c \in \{1, \infty\}$ . For all other pairs, we set  $c = 1$ .

### 4.3.2. Training & evaluation

DKGs were created by sampling based on strategies described in Section 4.2.1 with values  $p \in \{0.4, 0.6, 0.8\}$ . There were three types of training setups: *single graph* (using only the DKG without any links to external graph); *synthetic scenario* (linked with original GKG); *real-world scenario* (linked with distinct GKG). Due to the non-trivial nature of GKG selection in the real-world scenario, we tested two different options for the GKG. DKGs were sampled from WN18RR, FB15k-237, and WD50K. For FB15k-237 and WD50K, node sampling had to be restricted due to the high diversity of nodes in val/test (see Section 4.2.1). In the real-world scenario, the ConceptNet, FB15k-237, and WD50K were utilized as GKGs. Experiments conducted on a single graph or for a synthetic scenario were executed 3 times each with a distinct random seed for both the sampling procedure and the model initialization. However, due to the low standard deviation observed across the runs, we reduced the number of runs to one for the real-world scenario. We reported the mean and standard deviation values of the runs. We used standard metrics for the KG completion task, computed on DKG:  $Hits@10$ , Mean Rank ( $MR$ ) and Mean Reciprocal Rank ( $MRR$ ). During the hyperparameter search, the mean  $Hits@10$  on val was chosen as the benchmark to determine the optimal set of hyperparameters.

The implementation used PyKEEN (Ali et al., 2021) and PyTorch (Paszke et al., 2019) libraries. To maintain reproducibility, we utilized DVC (Kupriev et al., 2023) and Weights&Biases (Biewald, 2020).

## 5. Results

### 5.1. Synthetic scenario

Results from the synthetic scenario are presented in Table 1. For each combination of (dataset, sampling,  $p$ ), we recorded the efficacy of the trained model on a *single* graph and using the proposed framework (linked). The relative difference between them (*boost*) is a crucial metric that evaluates the framework. The results show that we can obtain significant improvement by linking GKG. Nevertheless, the boost is strictly dependent on the sampling setting. The more rigorous the sampling setting, the greater the increase in effectiveness (see Figure 3). The negative correlation between  $p$  and performance boost indicates that our framework can be helpful, especially in the harsh conditions of early stages of KG development or when it has limited data. Having only 40% of triples we achieved 44.9%/0.0%/16.7%  $Hits@10$  boost on WN18RR/FB15k-237/WD50K respectively. However, the effectiveness differs among the datasets. For instance, while WN18RR experienced a significant boost, FB15k-237 showed a smaller increase. This variability can be attributed to each dataset’s inherent characteristics and complexities.

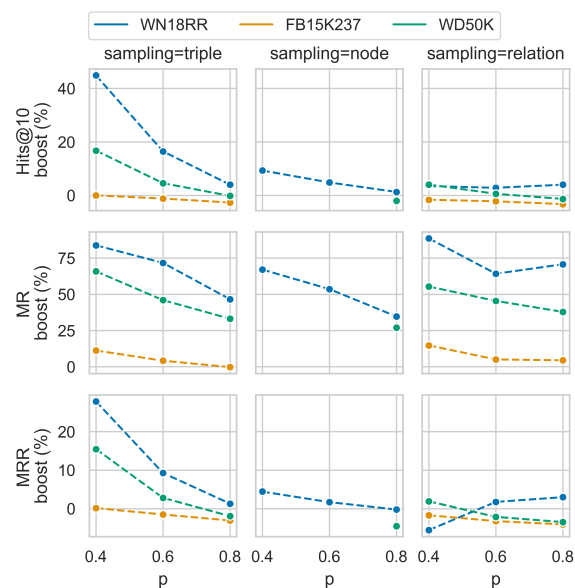


Figure 3: Synthetic scenario: performance boost across different sampling strategies and varying rates  $p$ . The boost is the performance improvement the framework achieves over a single graph. Please note that the relation sampling datasets should not be directly compared amongst themselves as the test was not preserved, causing high standard deviation (see 4.2.1).

sampling	$p$	$Hits@10 \uparrow$			$MR \downarrow$			$MRR \uparrow$		
		single	linked	boost(%)	single	linked	boost(%)	single	linked	boost(%)
<b>WN18RR</b>										
triple	0.4	0.347 ± 0.005	<b>0.502 ± 0.006</b>	44.9	7681 ± 62	<b>1245 ± 44</b>	83.8	0.270 ± 0.003	<b>0.345 ± 0.001</b>	27.8
triple	0.6	0.446 ± 0.001	<b>0.519 ± 0.004</b>	16.4	4908 ± 172	<b>1392 ± 15</b>	71.6	0.342 ± 0.003	<b>0.373 ± 0.004</b>	9.3
triple	0.8	0.525 ± 0.004	<b>0.546 ± 0.004</b>	4.0	2685 ± 155	<b>1435 ± 52</b>	46.6	0.416 ± 0.005	<b>0.421 ± 0.003</b>	1.3
node	0.4	0.546 ± 0.004	<b>0.597 ± 0.002</b>	9.3	2164 ± 52	<b>713 ± 18</b>	67.0	0.473 ± 0.002	<b>0.494 ± 0.001</b>	4.4
node	0.6	0.562 ± 0.000	<b>0.590 ± 0.001</b>	4.8	2044 ± 29	<b>950 ± 15</b>	53.6	0.480 ± 0.001	<b>0.488 ± 0.001</b>	1.7
node	0.8	0.576 ± 0.004	<b>0.583 ± 0.001</b>	1.3	1829 ± 8	<b>1194 ± 20</b>	34.7	<b>0.484 ± 0.001</b>	0.482 ± 0.001	-0.2
relation	0.4	0.696 ± 0.182	<b>0.721 ± 0.184</b>	3.5	1083 ± 450	<b>124 ± 70</b>	88.5	<b>0.590 ± 0.233</b>	0.557 ± 0.212	-5.6
relation	0.6	0.755 ± 0.129	<b>0.777 ± 0.143</b>	2.8	1542 ± 622	<b>552 ± 649</b>	64.2	0.684 ± 0.145	<b>0.696 ± 0.160</b>	1.7
relation	0.8	0.731 ± 0.115	<b>0.760 ± 0.135</b>	4.0	2013 ± 157	<b>591 ± 671</b>	70.7	0.659 ± 0.130	<b>0.678 ± 0.149</b>	3.0
Max boost (%)				44.9			88.5			27.8
Mean boost (%)				10.1			64.5			4.8
<b>FB15k-237</b>										
triple	0.4	<b>0.360 ± 0.002</b>	0.360 ± 0.000	0.0	315 ± 4	<b>279 ± 1</b>	11.2	0.204 ± 0.001	<b>0.204 ± 0.001</b>	0.1
triple	0.6	<b>0.398 ± 0.001</b>	0.393 ± 0.003	-1.2	242 ± 1	<b>231 ± 0</b>	4.2	<b>0.227 ± 0.001</b>	0.224 ± 0.002	-1.5
triple	0.8	<b>0.442 ± 0.000</b>	0.430 ± 0.002	-2.7	<b>194 ± 2</b>	195 ± 1	-0.3	<b>0.254 ± 0.001</b>	0.246 ± 0.000	-3.1
relation	0.4	<b>0.480 ± 0.009</b>	0.472 ± 0.010	-1.6	256 ± 68	<b>218 ± 61</b>	14.7	<b>0.297 ± 0.019</b>	0.292 ± 0.018	-1.7
relation	0.6	<b>0.490 ± 0.009</b>	0.479 ± 0.009	-2.2	197 ± 39	<b>187 ± 39</b>	5.0	<b>0.308 ± 0.002</b>	0.298 ± 0.000	-3.2
relation	0.8	<b>0.502 ± 0.014</b>	0.485 ± 0.011	-3.3	169 ± 17	<b>162 ± 12</b>	4.4	<b>0.304 ± 0.005</b>	0.291 ± 0.004	-4.1
Max boost (%)				0.0			14.7			0.1
Mean boost (%)				-1.8			6.5			-2.3
<b>WD50K</b>										
triple	0.4	0.283 ± 0.001	<b>0.330 ± 0.000</b>	16.7	1713 ± 42	<b>585 ± 3</b>	65.8	0.164 ± 0.001	<b>0.189 ± 0.001</b>	15.5
triple	0.6	0.353 ± 0.000	<b>0.369 ± 0.001</b>	4.5	931 ± 20	<b>503 ± 5</b>	45.9	0.208 ± 0.001	<b>0.213 ± 0.000</b>	2.8
triple	0.8	<b>0.400 ± 0.001</b>	0.399 ± 0.001	-0.2	667 ± 6	<b>446 ± 4</b>	33.1	<b>0.240 ± 0.001</b>	0.235 ± 0.000	-2.0
node	0.8	<b>0.440 ± 0.002</b>	0.431 ± 0.001	-2.0	520 ± 2	<b>380 ± 1</b>	27.0	<b>0.271 ± 0.001</b>	0.259 ± 0.002	-4.6
relation	0.4	0.413 ± 0.023	<b>0.430 ± 0.029</b>	4.0	768 ± 123	<b>343 ± 97</b>	55.3	0.249 ± 0.025	<b>0.254 ± 0.027</b>	1.9
relation	0.6	0.437 ± 0.014	<b>0.439 ± 0.007</b>	0.6	617 ± 42	<b>337 ± 64</b>	45.4	<b>0.269 ± 0.013</b>	0.263 ± 0.008	-2.2
relation	0.8	<b>0.447 ± 0.016</b>	0.441 ± 0.008	-1.3	541 ± 83	<b>336 ± 67</b>	37.8	<b>0.275 ± 0.010</b>	0.266 ± 0.005	-3.5
Max boost (%)				16.7			65.8			15.5
Mean boost (%)				3.2			44.4			1.1

Table 1: Synthetic scenario: performance comparison on KG completion task in both single and linked settings across various sampling types and rates  $p$ . Each result is presented as a mean value and standard deviation. The superior performance for each metric, sampling and  $p$  combination is highlighted. The boost is the performance improvement the framework achieves over a single graph. Please note that the results on relation sampling datasets should not be directly compared amongst themselves as the test was not preserved, causing high standard deviation (see 4.2.1).

The following observation emerges when comparing the effects of different sampling strategies on boost. Triple sampling yielded a significantly higher boost compared to node sampling. Interestingly, node sampling maintained a higher base performance on the single graph than achieved with triple sampling ( $Hits@10$  of 0.546 vs 0.347). This potentially explains the lower boost observed with node sampling. It indicates that while node sampling retains more integral information about the graph, triple sampling, being more disruptive, allows for a higher improvement using our framework.

In Figure 4, we can observe that while using node sampling, performance on a single graph does not decline significantly with lower  $p$ , and the potential for improvement is limited. Conversely, when sampling triples, there is a significant performance drop, and our framework could help to mitigate it. Therefore, increasing the number of nodes can lead to better outcomes than triples when expanding one’s KG. Results for other datasets also confirm this statement (see Appendix A: Figure 5, 6).

## 5.2. Real-world scenario

Results from experiments in the real-world scenario are presented in Table 2. We only show-cased DKGs created by sampling triples, as it yielded the most significant differences in efficacy on the single graph (see Section 5.1). This choice allows a more convenient analysis of the framework’s effectiveness. For each combination of (dataset,  $p$ ), we recorded the efficacy of the trained model on a single graph and utilised the proposed framework by linking it to external KG. We explored two GKGs as potential external sources for each dataset, given that finding a suitable match is not straightforward. Similar to the previous scenario, the crucial metric remains the boost, which again shows dependency on the  $p$  value.

Under the most rigorous conditions (40% of triples), we observed the highest efficacy improvement, with a 14.0%/ - 6.9%/10.8%  $Hits@10$  boost on WN18RR-ConceptNet, FB15k-237-YAGO3-10, and WD50K-FB15k-237 pairs, respectively. A significant success was on WN18RR and WD50K. This suggests that our framework can enrich the embeddings of small-scale KGs in real-life situations. However, using FB15k-237 or less strict set-

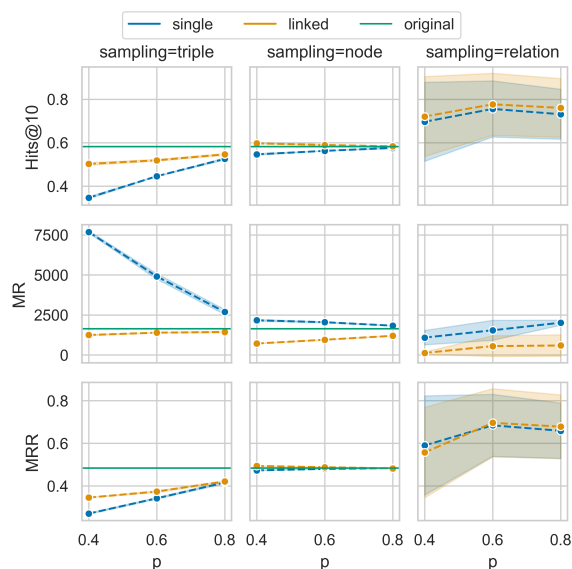


Figure 4: WN18RR: Performance across different sampling strategies and varying rates  $p$ . Two settings are shown: training on the single and linked graph. The green line shows the performance on the original graph ( $p = 1.0$ ). The shaded areas represent the standard deviation across multiple runs. Please note that the relation sampling datasets should not be directly compared amongst themselves as the test was not preserved, causing high standard deviation (see Section 4.2.1).

tings, the framework hurt performance. While our framework shows promise, performance improvement is not guaranteed in all situations. Further, in most cases, only one of two GKGs provided improvement. This confirms the mentioned requirement that the chosen GKG must be well-suited to the DKG.

### 5.3. Common findings

In both scenarios, we observed varying degrees of improvement across different metrics. The highest boost was observed for MR, followed by Hits@10, with MRR registering the most minor enhancement. Notably, in some cases, the boost for the MR metric was negative, even while positive improvements were noted for other metrics. We found that determining the optimal loss was not obvious, as there was a slight variance in mean performance on the downstream task. Nevertheless, insights from the ablation study indicated that the trained models predict lower scores for incorrect linking triples. On average, the score of incorrect links was lower by 5.00% while correct links' score only by 0.77% (see Appendix A: Figure 7). This indicates that the proposed loss can incorporate alignment probability into embeddings.

## 6. Limitations and Future Work

One identified limitation is that the method may encounter challenges when DKG and GKG have a small number of common entities. A potential future direction involves addressing this issue by enhancing the method's robustness to the absence of an entity in GKG.

Moreover, in the scenario of small, domain-specific KGs already containing all necessary information for the completion task, our method—by adding general knowledge—might not substantially enhance performance. Nevertheless, it holds potential benefits for tasks requiring more distant connections than KG completion.

## 7. Conclusion

In this paper, we proposed the framework for enriching embeddings of domain-specific KG by aligning and linking entities to general-purpose KG. The framework is general and modular, meaning we can use any alignment method, representation model and loss function. We utilized a simple alignment method based on the textual embeddings of entities and their neighbourhood. Notably, this alignment approach did not require additional annotation, making it readily applicable. Moreover, we proposed the weighted loss function, which could help mitigate entity alignment's negative effects.

We examined the proposed framework in extensive experiments, both in synthetic and real-world scenarios. The evaluation methodology was tailored to simulate the early stages of KG development, which are rigorous conditions where a KG may lack comprehensive information. Our results demonstrated that utilizing the knowledge from GKGs could significantly improve performance on the downstream task. However, it is essential to highlight that the degree of improvement depends on the particular domain-specific KG and linked general-purpose KG. This research underscores the possibilities of harnessing established GKGs to strengthen emerging KGs, offering a direction for improving their utility in real-world scenarios. Given that enriching small-scale KGs is an unexplored research area, this paper not only presents a unique approach but also signals a potential pathway for future research.

## 8. Acknowledgments

This work was funded by the European Union under the Horizon Europe grant OMINO – Overcoming Multilevel INformation Overload (grant number 101086321, <http://ominoproject.eu/>).



dataset	p	Hits@10 ↑				MR ↓				MRR ↑			
		single	linked		boost(%)	single	linked		boost(%)	single	linked		boost(%)
WN18RR	0.4	0.347 ± 0.005	<b>0.395</b>	0.338	14.0	7681 ± 62	<b>1508</b>	4050	80.4	<b>0.270 ± 0.003</b>	0.259	0.262	-3.2
WN18RR	0.6	0.446 ± 0.001	<b>0.471</b>	0.435	5.7	4908 ± 172	<b>963</b>	2685	80.4	<b>0.342 ± 0.003</b>	0.330	0.330	-3.3
WN18RR	0.8	0.525 ± 0.004	<b>0.527</b>	0.504	0.3	2685 ± 155	<b>750</b>	1725	72.1	<b>0.416 ± 0.005</b>	0.390	0.400	-3.8
Max boost (%)					14.0				80.4				-3.2
Mean boost (%)					6.7				77.6				-3.5
FB15K237	0.4	<b>0.360 ± 0.002</b>	0.330	Y3-10	-6.9	315 ± 4	CN	Y3-10	1.4	<b>0.204 ± 0.001</b>	CN	Y3-10	-5.6
FB15K237	0.6	<b>0.398 ± 0.001</b>	0.359	0.362	-9.1	242 ± 1	315	<b>310</b>	-3.9	<b>0.227 ± 0.001</b>	0.189	0.192	-8.6
FB15K237	0.8	<b>0.442 ± 0.000</b>	0.387	0.395	-10.6	194 ± 2	254	251	-9.5	<b>0.254 ± 0.001</b>	0.202	0.208	-11.5
Max boost (%)					-6.9				1.4				-5.6
Mean boost (%)					-8.9				-4.0				-8.6
WD50K	0.4	0.283 ± 0.001	<b>0.313</b>	FB	10.8	1713 ± 42	FB	Y3-10	55.8	0.164 ± 0.001	<b>0.181</b>	0.163	10.6
WD50K	0.6	0.353 ± 0.000	<b>0.361</b>	0.327	2.4	931 ± 20	<b>558</b>	644	40.0	0.208 ± 0.001	<b>0.212</b>	0.190	2.0
WD50K	0.8	<b>0.400 ± 0.001</b>	0.396	0.358	-0.9	667 ± 6	<b>450</b>	510	32.6	<b>0.240 ± 0.001</b>	0.236	0.210	-1.8
Max boost (%)					10.8				55.8				10.6
Mean boost (%)					4.1				42.8				3.6

Table 2: Real-world scenario: performance comparison on KG completion task in both single and linked settings across various datasets sampled using triple sampling and rates  $p$ . The DSKGs were linked to two of the following GKGs: ConceptNet (CN), FB15k-237 (FB) and YAGO3-10(Y3-10). Each result is presented as a mean value and standard deviation, where applicable. The superior performance for each metric, dataset and  $p$  combination is highlighted in bold. The boost is the performance improvement of linking a GKG that yielded the higher score over training on a single graph.

Views and opinions expressed are those of the authors alone and do not necessarily reflect those of the European Union or the European Research Executive Agency. Neither the European Union nor the European Research Executive Agency can be held responsible for them. It was also co-financed with funds from the Polish Ministry of Education and Science under the programme entitled International Co-Financed Projects, grant no. 573977. This work was also funded in part by the National Science Centre, Poland under CHIST-ERA Open & Re-usable Research Data & Software (grant number 2022/04/Y/ST6/00183) as well Department of Artificial Intelligence, Wroclaw Tech.

## 9. Bibliographical References

- Ralph Abboud, İsmail İlkan Ceylan, Thomas Lukasiewicz, and Tommaso Salvatori. 2020. BoxE: A box embedding model for knowledge base completion. In *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS'20*, pages 9649–9661, Red Hook, NY, USA. Curran Associates Inc.
- Vaibhav Adlakha, Parishad BehnamGhader, Xing Han Lu, Nicholas Meade, and Siva Reddy. 2023. [Evaluating correctness and faithfulness of instruction-following models for question answering](#).
- 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):8825–8845.
- Mehdi Ali, Max Berrendorf, Charles Tapley Hoyt, Laurent Vermue, Sahand Sharifzadeh, Volker Tresp, and Jens Lehmann. 2021. [PyKEEN 1.0: A python library for training and evaluating knowledge graph embeddings](#). *Journal of Machine Learning Research*, 22(82):1–6.
- Ivana Balazevic, Carl Allen, and Timothy Hospedales. 2019. [Tucker: Tensor factorization for knowledge graph completion](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics.
- Y. Bengio, Aaron Courville, and Pascal Vincent. 2013. [Representation learning: A review and new perspectives](#). In *IEEE transactions on pattern analysis and machine intelligence*, volume 35, pages 1798–1828.
- Tim Berners-Lee et al. 1998. Semantic web road map.
- Lukas Biewald. 2020. [Experiment tracking with](#)

- [weights and biases](#). Software available from wandb.com.
- Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. [Translating Embeddings for Modeling Multi-relational Data](#). In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.
- Samuel R. Bowman. 2023. [Eight things to know about large language models](#).
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners.
- Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. 2017. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *Proceedings of the 26th international joint conference on artificial intelligence, IJCAI'17*, pages 1511–1517. AAAI Press.
- Pedro Colon-Hernandez, Catherine Havasi, Jason Alonso, Matthew Huggins, and Cynthia Breazeal. 2021. [Combining pre-trained language models and structured knowledge](#).
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. 2017. Neural message passing for quantum chemistry. In *Proceedings of the 34th international conference on machine learning - volume 70, ICML'17*, pages 1263–1272, Sydney, NSW, Australia. JMLR.org. Number of pages: 10.
- Thomas R Gruber. 1995. Toward principles for the design of ontologies used for knowledge sharing? *International journal of human-computer studies*, 43(5-6):907–928.
- William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. [Representation learning on graphs: Methods and applications](#). In *IEEE Data Eng. Bull.*, volume 40, pages 52–74.
- Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. 2022. [A Survey on Knowledge Graphs: Representation, Acquisition, and Applications](#). In *IEEE Transactions on Neural Networks and Learning Systems*, volume 33, pages 494–514. Conference Name: IEEE Transactions on Neural Networks and Learning Systems.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.
- Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. [Bag of tricks for efficient text classification](#).
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International conference on learning representations (ICLR)*.
- Ruslan Kuprieiev, Skshetry, Peter Rowlands ( ), Dmitry Petrov, Paweł Redzyński, Casper Da Costa-Luis, David De La Iglesia Castro, Alexander Schepanovski, Ivan Shcheklein, , Gao, Batuhan Taskaya, Jorge Orpinel, Dave Berenbaum, Fábio Santos, , Daniele, Ronan Lamy, Aman Sharma, Zhanibek Kaimuldenov, Dani Hodovic, Nikita Kodenko, Andrew Grigorev, , Earl, Nabanita Dash, George Vyshnya, Maykulkarni, Max Hora, , Vera, and Sanidhya Mangal. 2023. [Dvc: Data version control - git for data & models](#).
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. [Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.
- Xiao Liu, Haoyun Hong, Xinghao Wang, Zeyi Chen, Evgeny Kharlamov, Yuxiao Dong, and Jie Tang. 2022. [Selfkg: Self-supervised entity alignment in knowledge graphs](#). *CoRR*, abs/2203.01044.
- Deepak Nathani, Jatin Chauhan, Charu Sharma, and Manohar Kaul. 2019. [Learning Attention-based Embeddings for Relation Prediction in Knowledge Graphs](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4710–4723, Florence, Italy. Association for Computational Linguistics.

- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. [PyTorch: An imperative style, high-performance deep learning library](#). In *Advances in neural information processing systems*, volume 32. Curran Associates, Inc.
- Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. [KILT: a Benchmark for Knowledge Intensive Language Tasks](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.
- Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. [Language models as knowledge bases?](#)
- Jay Pujara, Eriq Augustine, and Lise Getoor. 2017. [Sparsity and Noise: Where Knowledge Graph Embeddings Fall Short](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1751–1756, Copenhagen, Denmark. Association for Computational Linguistics.
- Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. [Modeling Relational Data with Graph Convolutional Networks](#). In *The Semantic Web*, Lecture Notes in Computer Science, pages 593–607, Cham. Springer International Publishing.
- Juan Sequeda and Ora Lassila. 2021. [Designing and Building Enterprise Knowledge Graphs](#). Springer International Publishing.
- Zequn Sun, Wei Hu, and Chengkai Li. 2017. Cross-lingual entity alignment via joint attribute-preserving embedding. In *ISWC*, pages 628–644.
- Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. [RotatE: Knowledge graph embedding by relational rotation in complex space](#). In *International conference on learning representations*.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. [Llama: Open and efficient foundation language models](#).
- Bayu Distiawan Trisedya, Jianzhong Qi, and Rui Zhang. 2019. [Entity Alignment between Knowledge Graphs Using Attribute Embeddings](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):297–304.
- Théo Trouillon, Christopher R. Dance, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2017. [Knowledge graph completion via complex tensor factorization](#).
- Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. 2019. Composition-based Multi-Relational Graph Convolutional Networks. In *International Conference on Learning Representations*.
- Kaisheng Zeng, Chengjiang Li, Lei Hou, Juanzi Li, and Ling Feng. 2021. [A comprehensive survey of entity alignment for knowledge graphs](#). *AI Open*, 2:1–13.
- Rui Zhang, Bayu Distiawan Trisedya, Miao Li, Yong Jiang, and Jianzhong Qi. 2022. [A benchmark and comprehensive survey on knowledge graph entity alignment via representation learning](#).
- Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. 2019. Quaternion knowledge graph embeddings. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 246, pages 2735–2745. Curran Associates Inc., Red Hook, NY, USA.

## 10. Language Resource References

- Dettmers, Tim and Minervini, Pasquale and Stenortorp, Pontus and Riedel, Sebastian. 2018. [Convolutional 2D knowledge graph embeddings](#). AAAI Press, AAAI'18/IAAI'18/EAAI'18.
- Galkin, Mikhail and Trivedi, Priyansh and Maheshwari, Gaurav and Usbeck, Ricardo and Lehmann, Jens. 2020. [Message Passing for Hyper-Relational Knowledge Graphs](#). Association for Computational Linguistics.
- Mahdisoltani, Farzaneh and Biega, Joanna and Suchanek, Fabian M. 2015. [YAGO3: A](#)

*Knowledge Base from Multilingual Wikipedias.*  
www.cidrdb.org.

Miller, George A. 1995. *WordNet: a lexical database for English.* ACM New York, NY, USA.

Speer, Robyn and Chin, Joshua and Havasi, Catherine. 2017. *ConceptNet 5.5: An open multilingual graph of general knowledge.* AAAI Press, AAAI'17. Place: San Francisco, California, USA Number of pages: 8.

Toutanova, Kristina and Chen, Danqi. 2015. *Observed versus latent features for knowledge base and text inference.* Association for Computational Linguistics.

Vrandečić, Denny and Krötzsch, Markus. 2014. *Wikidata: a free collaborative knowledgebase.* ACM New York, NY, USA.

## A. Supplementary results

We extend the results of the synthetic scenario discussed in Section 5.1 by providing performance analysis across different sampling strategies on Figures 5 and 6 for datasets FB15k-237 and WD50K, respectively.

We further explore the common findings outlined in Section 5.3 by conducting a comparative analysis of predicted scores for linking triples. This analysis contrasts models trained using the proposed loss function against those trained with the standard loss function.

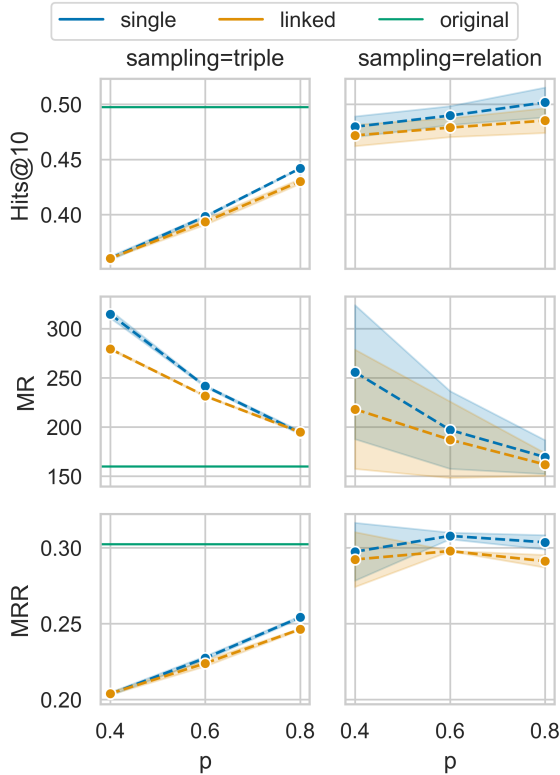


Figure 5: FB15k-237: Performance across different sampling strategies and varying rates  $p$ . Two settings are shown: training on the single and linked graph. The green line shows the performance on the original graph ( $p = 1.0$ ). The shaded areas represent the standard deviation across multiple runs. Please note that the relation sampling datasets should not be directly compared amongst themselves as the test was not preserved, causing high standard deviation (see Section 4.2.1).

## B. Supplementary statistics of datasets

We provide detailed statistics of the original and sampled datasets in Tables 3 and 4, respectively.

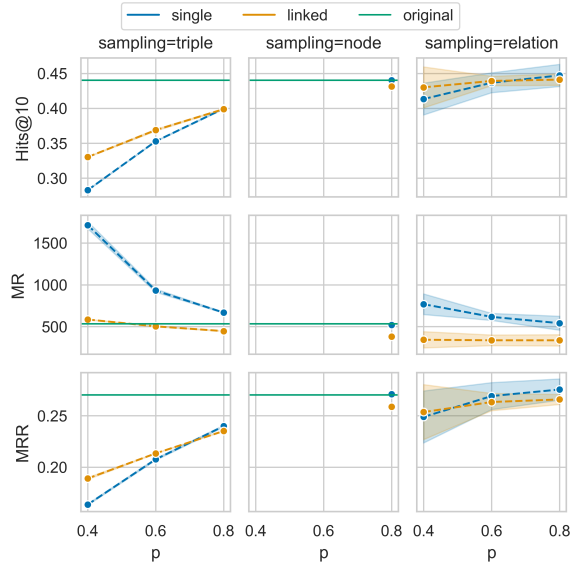


Figure 6: WD50K: Performance across different sampling strategies and varying rates  $p$ . Two settings are shown: training on the single and linked graph. The green line shows the performance on the original graph ( $p = 1.0$ ). The shaded areas represent the standard deviation across multiple runs. Please note that the relation sampling datasets should not be directly compared amongst themselves as the test was not preserved, causing high standard deviation (see Section 4.2.1).

## C. Computational complexity

First, denote  $n$  and  $o$  as the number of DKG triples and entities, respectively. RotatE time complexity is  $O(n)$ , and space complexity is  $O(o)$ . Further, in our method we add GKG with  $m$  triples and  $p$  entities, linking each entity of DKG with  $k$  nearest neighbors in GKG. Assuming the same embedding dimension, the time complexity of the proposed method is  $O(n + m + k * o)$ , with space complexity  $O(o + p)$ .

## D. Alignment: entity representation

Based on the preliminary results, we selected the best-performing method for entity representation, described in Section 3.2. In this section, we provide an ablation study.

The proposed concatenation method yielded promising results, achieving, on average, a 10% higher accuracy during entity alignment compared to the absence of concatenation (0.901 vs. 0.819). Without concatenation, we averaged embeddings across entity labels, outgoing neighbors' labels, and ingoing neighbors' labels. This experiment was conducted on all pairs of datasets utilized in

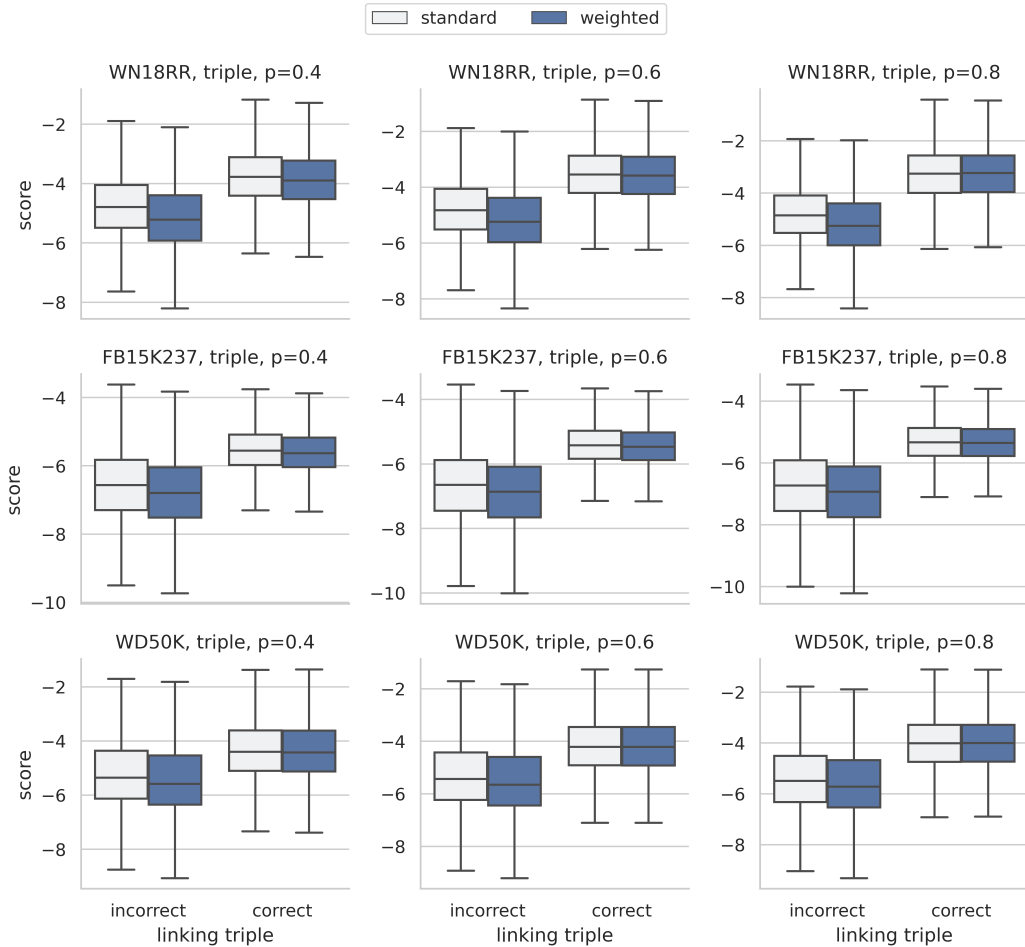


Figure 7: Comparative analysis of predicted scores for incorrect and correct linking triples from the training set for each setting (dataset, sampling,  $p$ ).

dataset	Triples			Entities			Relations		
	train	val	test	train	val	test	train	val	test
WN18RR*	86 835	2817	2923	40 714	4835	4985	11	11	11
FB15k-237*†	272 115	17 526	20 438	14 505	9799	10 317	237	237	237
WD50K*	164 631	22 429	45 284	40 107	16 500	22 732	473	297	347
ConceptNet†	3 423 004	0	0	1 787 373	0	0	47	47	47
YAGO3-10†	1 079 040	4978	4982	123 143	7914	7906	37	37	37

Table 3: Statistics of the original datasets used in our experiments. The table shows the number of triples, entities, and relations across the training (train), validation (val), and test sets for each dataset. Statistics refer to original datasets. Some were sampled or cropped during experiments (see Sections 4.2.1 and 4.3.1). Datasets used as DSKG are marked with \*, while datasets used as GKG are marked with †.

the synthetic scenario that allows for such evaluation.

sampling p	Triples			Entities			Relations			
	train	val	test	train	val	test	train	val	test	
<b>WN18RR</b>										
triple	0.4	34 734 ± 0	2817 ± 0	2923 ± 0	30 975 ± 57	4835 ± 0	4985 ± 0	11 ± 0	11 ± 0	11 ± 0
triple	0.6	52 101 ± 0	2817 ± 0	2923 ± 0	36 178 ± 27	4835 ± 0	4985 ± 0	11 ± 0	11 ± 0	11 ± 0
triple	0.8	69 468 ± 0	2817 ± 0	2923 ± 0	39 093 ± 28	4835 ± 0	4985 ± 0	11 ± 0	11 ± 0	11 ± 0
node	0.4	28 815 ± 78	2817 ± 0	2923 ± 0	16 286 ± 0	4835 ± 0	4985 ± 0	11 ± 0	11 ± 0	11 ± 0
node	0.6	46 715 ± 90	2817 ± 0	2923 ± 0	24 428 ± 0	4835 ± 0	4985 ± 0	11 ± 0	11 ± 0	11 ± 0
node	0.8	66 311 ± 157	2817 ± 0	2923 ± 0	32 571 ± 0	4835 ± 0	4985 ± 0	11 ± 0	11 ± 0	11 ± 0
relation	0.4	20 367 ± 18 482	511 ± 625	545 ± 616	14 485 ± 10 417	933 ± 1168	989 ± 1150	4 ± 0	4 ± 0	4 ± 0
relation	0.6	57 790 ± 19 777	1765 ± 624	1827 ± 635	32 006 ± 7832	3135 ± 1000	3228 ± 1025	7 ± 0	7 ± 0	7 ± 0
relation	0.8	61 586 ± 17 153	1852 ± 598	1902 ± 610	34 809 ± 4855	3286 ± 949	3353 ± 974	9 ± 0	9 ± 0	9 ± 0
<b>FB15k-237</b>										
triple	0.4	108 846 ± 0	17 526 ± 0	20 438 ± 0	14 187 ± 10	9799 ± 0	10 317 ± 0	237 ± 0	223 ± 0	224 ± 0
triple	0.6	163 269 ± 0	17 526 ± 0	20 438 ± 0	14 352 ± 6	9799 ± 0	10 317 ± 0	237 ± 0	223 ± 0	224 ± 0
triple	0.8	217 692 ± 0	17 526 ± 0	20 438 ± 0	14 439 ± 7	9799 ± 0	10 317 ± 0	237 ± 0	223 ± 0	224 ± 0
relation	0.4	103 999 ± 16 860	6364 ± 891	7419 ± 1056	12 884 ± 220	5457 ± 707	5981 ± 722	95 ± 0	87 ± 2	89 ± 3
relation	0.6	156 537 ± 20 322	9562 ± 1011	11 186 ± 1268	13 779 ± 352	7218 ± 571	7747 ± 589	142 ± 0	133 ± 3	134 ± 3
relation	0.8	224 750 ± 25 236	14 309 ± 1632	16 748 ± 1881	14 332 ± 75	8725 ± 633	9291 ± 618	190 ± 0	178 ± 2	180 ± 2
<b>WD50K</b>										
triple	0.4	65 852 ± 0	22 429 ± 0	45 284 ± 0	34 164 ± 32	16 500 ± 0	22 732 ± 0	434 ± 2	297 ± 0	347 ± 0
triple	0.6	98 779 ± 0	22 429 ± 0	45 284 ± 0	37 091 ± 41	16 500 ± 0	22 732 ± 0	453 ± 6	297 ± 0	347 ± 0
triple	0.8	131 705 ± 0	22 429 ± 0	45 284 ± 0	38 957 ± 16	16 500 ± 0	22 732 ± 0	464 ± 3	297 ± 0	347 ± 0
node	0.8	147 774 ± 42	22 429 ± 0	45 284 ± 0	32 086 ± 0	16 500 ± 0	22 732 ± 0	450 ± 1	297 ± 0	347 ± 0
relation	0.4	62 701 ± 11 210	7865 ± 1591	16 014 ± 3259	25 204 ± 2662	7700 ± 956	11 535 ± 1186	189 ± 0	112 ± 6	127 ± 5
relation	0.6	88 487 ± 12 893	11 529 ± 1838	23 332 ± 3923	29 090 ± 3143	10 258 ± 1183	14 848 ± 1580	284 ± 0	173 ± 5	201 ± 6
relation	0.8	121 499 ± 15 858	16 250 ± 2317	32 947 ± 4969	33 181 ± 2970	12 848 ± 1161	18 053 ± 1489	378 ± 0	238 ± 4	275 ± 5

Table 4: Statistics of the sampled datasets used in our experiments. The table shows the number of triples, entities, and relations across the training (train), validation (val), and test sets for each dataset.