Dynamic Knowledge Graph Embeddings via Local Embedding Reconstructions

Franz Krause[0000−0003−2869−6815]

Data and Web Science Group, University of Mannheim
Mannheim, Germany
franz.krause@uni-mannheim.de

Abstract. Knowledge graph embeddings and successive machine learning models represent a topic that has been gaining popularity in recent research. These allow the use of graph-structured data for applications that, by definition, rely on numerical feature vectors as inputs. In this context, the transformation of knowledge graphs into sets of numerical feature vectors is performed by embedding algorithms, which map the elements of the graph into a low-dimensional embedding space. However, these methods mostly assume a static knowledge graph, so subsequent updates inevitably require a re-run of the embedding process. In this work the **Navi Approach** is introduced which aims to maintain advantages of established embedding methods while making them accessible to dynamic domains. Relational Graph Convolutional Networks are adapted for reconstructing node embeddings based solely on local neighborhoods. Moreover, the approach is independent of the original embedding process, as it only considers its resulting embeddings. Preliminary results suggest that the performance of successive machine learning tasks is at least maintained without the need of relearning the embeddings nor the machine learning models. Often, using the reconstructed embeddings instead of the original ones even leads to an increase in performance.

Keywords: Knowledge Graph · Semantic Web · Dynamic Embeddings

1 Introduction and Motivation

Knowledge graphs (KGs) have emerged as an effective tool for managing semi-structured domain knowledge so that it can be made available in a human- and machine-interpretable way. Their inherent information is encoded as triples $(subject, predicate, object)$ and used to improve the performance in areas such as question answering [5] and recommendation [9]. Moreover, KGs play a key role in data-intensive domains like Industry 4.0, where they have so far mostly been used for interconnecting technologies to improve efficiency and productivity of existing processes [2]. In order to make the information encoded within the knowledge graph available for machine learning models, representational learning methods are used to map its entities into a low-dimensional numerical space.

Category: Early Stage PhD

These representations of the entities, so-called embeddings, can then be processed by successive machine learning models. In the context of Industry 4.0, for example, embeddings are increasingly used as inputs for domain applications. Furthermore, current developments in this field, such as the transition to Industry 5.0, focus on human-machine collaboration, making a well-defined communication and interaction a crucial component for promoting trust [7]. The relevance of knowledge graphs as common specifications of a shared conceptualization will therefore increase even further.
Besides that, accumulation of interconnected devices in this context, as well as in other domains, is causing an increase in updates to KGs in the form of additions or deletions of edges and nodes. A knowledge graph should therefore be considered as dynamic, i.e., its topological structure or attributive information is changing over time. However, current methods lack this kind of flexibility. Usually, static KGs are assumed whose encoded information changes little or not at all. As soon as the graph is updated, the embeddings have to be relearned for the whole KG in order to include the updated information. For large-scale graphs, this relearning can be very time and resource intensive. Furthermore, retraining usually provides structurally similar but not identical embeddings for existing nodes, which is why successive machine learning models have to be adapted as well. Therefore, due to the increasing dynamics and scope of prospective KGs, these methods cannot be considered a suitable solution.
In this PhD, the Navi Approach is introduced to fill this gap towards the application of KG embeddings in dynamic domains. State of the art embedding algorithms for static knowledge graphs are solely used as blueprints for simplified and local reconstructions. It is shown that in this way established embedding methods are adapted such that dynamics are enabled as well.

2 State of the Art

Static Knowledge Graph Embeddings. In order to be able to use contextual information from a knowledge graph, embedding methods are usually considered, which assign numerical feature vectors to the nodes based on the topology of the graph. In this context, Relational Graph Convolutional Networks (R-GCNs) [15] accumulate the feature vectors of neighboring nodes based on the local structure of a KG. However, R-GCNs are inevitably customized to a specific use case and usually not all nodes in the graph are assigned an embedding. In contrast, RDF2Vec [11] generates all-embracing embeddings for different data mining tasks [20]. However, as a transductive method, the embeddings must always be relearned when the graph is updated. ConvE [4], as being another neural network based approach, uses two-dimensional embeddings as input, as well as convolutional layers and fully connected layers to reduce the number of parameters. It has been shown that extensions of TransE [3] are capable of embedding symmetric, asymmetric, inverse and compositional relations, though the basic idea of using a translation-based scoring function is preserved in all of them [16]. In contrast, compositional models apply tensor factorizations

to compute the embeddings [8, 23] and extensions to the compositional models employ complex numbers to be able to model binary relations [19]. Moreover, embedding methods exist which also take literals into account [13]. However, the embedding methods mentioned so far are trained for an entire static knowledge graph at once. Thus, possible dynamics of a graph in the form of additions and deletions of edges are not considered, which often impairs their use in practice.
Dynamic Knowledge Graph Embeddings. As an adaptation of the static embedding method TransE, puTransE [17] allows parallelization and orchestration across multiple machines and is thus capable of handling the dynamics in KGs. Another of the few approaches to dynamic embeddings is represented by DKGE [22], which is also based on TransE. Here, dynamics are enabled by restricting the retraining of the embeddings to the context of updated nodes, i.e., their neighboring nodes. Thus, the integration of new information is accelerated compared to full relearning. In the area of dynamic graphs in general, methods employ deep recurrent architectures [18], as well as self-attention layers [14] which encode temporally evolving structural information. However, none of the existing approaches is general enough to handle arbitrary embedding methods. In addition, updates in the graph necessarily require a relearning of embeddings, even if only for subgraphs, which in turn affects the overall structure of the embeddings and therefore also successive machine learning applications.

3 Problem Statement and Contributions

Based on the motivation and the state of the art in learning representations for knowledge graphs, there is a gap that we intend to fill in this PhD. We plan to provide a method that reuses existing embeddings, even if they assume static knowledge graphs, to make them usable for dynamic knowledge graphs as well. Thus, we formulate the following hypothesis with the related research questions.

Hypothesis. Static embeddings for a knowledge graph $\mathcal{G}$ can serve as input of a generalized and simplified method that incorporates the neighborhood of a node in order to derive its corresponding embedding independently of itself, thus enabling the generation of dynamic embeddings.

- **RQ1.** Can static embedding methods be adapted to obtain dynamic embeddings for new or updated nodes without relearning the original model?
- **RQ2.** What impact do knowledge graph updates, such as the addition of entities and removal of edges, have on the performance of generated embeddings with respect to entity classification and link prediction?

With this hypothesis and the related research questions, we intend to contribute to the Semantic Web community by developing a generalized method that leverages any numerical embedding by reconstructing it based on local neighborhoods to provide simplified ad-hoc embeddings for dynamic KGs. To the best of our knowledge, such an approach does not exist yet and will have a high impact on the Semantic Web community, as well as in dynamic domains like Industry 4.0.

The first research question RQ1 aims at elaborating methods for the desired dynamic embeddings based on static embedding algorithms. We propose to justify the restriction to local reconstructions within our approach via the so-called message passing paradigm, which states that existing embeddings are derivable from edges to neighboring nodes and their embeddings. It is to be examined whether this paradigm may be assumed for arbitrary embeddings, i.e., whether local topologies are always of relevance for embedding generations.
In RQ2, we further investigate the performance of our reconstruction approach on machine learning tasks with respect to different modifications of the KG. We will also study the performance of dynamic embeddings regarding the amount of modification to a KG. Furthermore, assuming the message passing paradigm, we need to investigate the impact of embedding reconstructions on further nodes, i.e., whether existing embeddings need to be adapted as well.

4 Research Methodology and Approach

The research of this PhD aims to define a generalized approach to the dynamization and simplification of static embeddings $\epsilon : V \mapsto \mathbb{R}^d$ with respect to a KG $\mathcal{G} = (V, E)$ which consists of the sets of nodes V and edges $E \subseteq V \times \mathcal{R} \times V$, where $\mathcal{R}$ denotes the set of valid entity relations. Thus, analogous to existing embeddings, no further information such as temporal edge properties are considered. The presented approach attempts to reconstruct the embedding $\epsilon(s)$ of a node $s \in V$ independently of $\epsilon(s)$ itself. Rather, it derives $\epsilon(s)$ from the neighborhood multiset $N(s)$ with elements from $V \setminus \{s\}$ such that distinct edges with adjacent nodes are taken into account as well. Due to the independence of the reconstruction and $\epsilon(s)$, self-loops are ignored but can be applied consecutively. The reconstructions $\Phi(s) \approx \epsilon(s)$ proposed in this paper are always structured as a composition $\Phi(s) = \phi(s) + \delta^*_\phi(s)$ of a deterministic ground assumption $\phi(s) \in \mathbb{R}^d$ and a trainable refinement term $\delta^*_\phi(s) \in \mathbb{R}^d$. By defining $\delta_\phi(s) := \epsilon(s) - \phi(s)$ as the noise term, the reconstruction error to be minimized is determined as

$$\epsilon(s) - \Phi(s) = \phi(s) + \delta_\phi(s) - \left[\phi(s) + \delta^*_\phi(s)\right] = \delta_\phi(s) - \delta^*_\phi(s).$$

By inserting this error into a suitable loss function $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}^+_0$ and applying a backpropagation method such as Adam [6], the reconstructed embedding $\Phi(s)$ is approximated to the target $\epsilon(s)$. In the following, analogous to [15], for each relation $r \in \mathcal{R}$ an inverse relation r' is assumed, such that each edge $(s, r, o) \in E$ implies an inverse edge (o, r', s) to simulate the varying influence of relations on subject and object nodes. Without loss of generality, the relations $\widehat{\mathcal{R}} = \bigcup_{r \in \mathcal{R}} \{r, r'\}$ are assumed in the following. Furthermore, $N_r(s) = \left\{v \in V \middle| \exists e \in E : e = (v, r, s)\right\}$ is defined as the set of all parent nodes of $s \in V$ with respect to $r \in \widehat{\mathcal{R}}$. This implies the neighborhood multiset $N(s) = \bigcup_{r \in \widehat{\mathcal{R}}} N_r(s)$, consisting of all parent and child nodes of s. As of now, reconstruction mappings $\Phi(\cdot)$ based on $N(s)$ are also referred to as **Navi Layers**,

like the **Connectivity Navi Layer** $\Phi_C(\cdot)$, defined by

$$\epsilon(s) \approx \Phi_C(s) = \frac{1}{|N(s)|}\left(\sum_{r\in\widehat{\mathcal{R}}}\sum_{y\in N_r(s)}\left[W_r + Id\right]\cdot\epsilon(y) + b_r\right) \tag{1}$$

with weight matrices $W_r \in \mathbb{R}^{d\times d}$ and bias terms $b_r \in \mathbb{R}^d$ for all relations $r \in \widehat{\mathcal{R}}$. With respect to the parameter initializations $W_r = 0$ and $b_r = 0$, the ground assumption $\phi_C(s) = \frac{1}{|N(s)|}\sum_{y\in N(s)}\epsilon(y)$ thus follows that the centroid of the neighbor embeddings should be close to the destined embedding $\epsilon(s)$. Finally, the refinement is achieved by training the weight matrices W_r and biases b_r in

$$\delta_{\phi_C^*}(s) = \frac{1}{|N(s)|}\left(\sum_{r\in\widehat{\mathcal{R}}}\sum_{y\in N_r(s)} W_r\cdot\epsilon(y) + b_r)\right)$$

for all relations $r \in \widehat{\mathcal{R}}$ such that the loss $\mathcal{L}\big(\epsilon\left(s\right) - \Phi_C\left(s\right)\big)$ is minimized. In particular, this approach resembles the idea of RDF2Vec embeddings, since the ground assumption does not include any information about the specific relations and considers connectedness as a cause of similarity. In contrast, the **Translational Navi Layer** $\Phi_T(\cdot)$ takes these relations into account by adapting translational approaches like TransE. These embed triples $(s, r, o) \in E$ such that $\epsilon(o) \approx \epsilon(s) + h(r)$ holds, where $h(r) \in \mathbb{R}^d$ is the embedding of relation r. By introducing the set $E_r := \{(\cdot, r, \cdot) \in E\}$ and the deterministic approximations $h_r := \frac{1}{|E_r|}\sum_{(s,r,o)\in E_r}\left[\epsilon(o) - \epsilon(s)\right] \approx h(r)$, the Navi Layer $\Phi_T(\cdot)$ is defined by

$$\epsilon(s) \approx \Phi_T(s) = \frac{1}{|N(s)|}\left(\sum_{r\in\widehat{\mathcal{R}}}\sum_{y\in N_r(s)}\left[W_r + Id\right]\cdot\left[\epsilon(y) + h_r\right] + b_r\right), \tag{2}$$

where the ground assumption and the refinement are defined analogously to $\Phi_C(\cdot)$, considering the transformations $\epsilon(y) \mapsto \epsilon(y) + h_r$.
The Navi Layers $\Phi_C(\cdot)$ and $\Phi_T(\cdot)$ can be interpreted as special cases of R-GCN layers according to [15] with the corresponding forward pass from layer l to $l+1$

$$h_s^{(l+1)} = \sigma\left(\left[W_0^{(l)}\cdot h_s^{(l)} + b_0^{(l)}\right] + \sum_{r\in\widehat{\mathcal{R}}}\sum_{y\in N_r(s)}\left[\frac{1}{c_{s,r}}W_r^{(l)}\cdot h_y^{(l)} + b_r^{(l)}\right]\right).$$

By specifying $h_s^{(l+1)} := \epsilon(s)$ and $h_y^{(l)} := \epsilon(y)$ or $h_y^{(l)} := \epsilon(y) + h_r$ with respect to the corresponding relation r, one obtains equations similar to (1) and (2). The required independence of $h_s^{(l+1)}$ and $h_s^{(l)}$ is preserved by defining $W_0^{(l)}$ and $b_0^{(l)}$ as deterministic zero elements. Finally, setting the activation function σ as the identity function and the normalization term $c_{s,r} := |N(s)|$ independent of relation r yields the forward passes of the Navi Layers $\Phi_C(\cdot)$ and $\Phi_T(\cdot)$, respectively. In addition, composite reconstructions of multiple Navi Layers are enabled via so-called **Navi Mergers**. These are defined as mappings $\gamma : \mathbb{R}^{d\times n} \to \mathbb{R}^d$, where n is the number of Navis Layers. A possible Navi Merger is the weighted mean

$$\gamma^{(wm)}\left(\Phi_1(s), ..., \Phi_n(s)\right) =: \gamma_n^{(wm)}(s) = \frac{1}{n}\sum_{i=1}^n W_i\cdot\Phi_i(s) + b_i,$$

with the trainable weight matrices $W_i \in \mathbb{R}^{d\times d}$ and biases $b_i \in \mathbb{R}^d$. In the following we denote a **Navi Approach/Reconstruction** $\gamma\left(\cdot\right) := \gamma\left(\Phi_1\left(\cdot\right), ..., \Phi_n\left(\cdot\right)\right)$ as a combination of $n \in \mathbb{N}$ Navi Layers $\Phi_1, ..., \Phi_n$ and a Navi Merger γ. The associated training is performed as an end-to-end procedure and the set of reconstructed embeddings for a KG $\mathcal{G} = (V, E)$ is denoted as $\Gamma = \{\gamma(v) : v \in V\}$.

5 Evaluation Plan

The evaluation of the Navi Approach is supposed to clarify whether and to what extent it answers the research questions RQ1 and RQ2 from Section 3. To assess the dynamics of Navi Reconstructions Γ, common benchmark methods like link prediction and entity classification are used. The latter is explained in the following as an example for the evaluation procedure depicted in Figure 1. However, the setting can be used almost analogously for link predictions, which take into account the deletion and addition of edges.

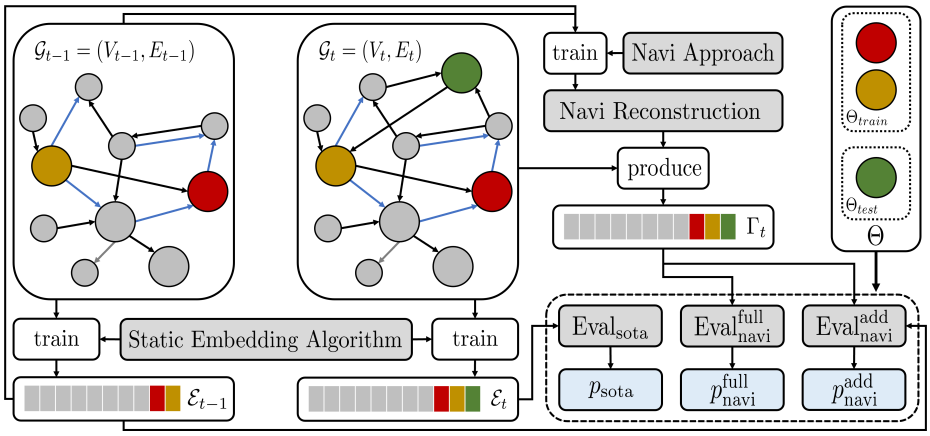

Fig. 1. Architecture of the Navi Reconstruction approach including the evaluation setup for simulating dynamic knowledge graphs.

The knowledge graphs AIFB, MUTAG, BGS and AM from [12] are particularly suitable as evaluation data as they each contain labeled nodes $\Theta \subset V$ including a split into train and test nodes $\Theta_{train}, \Theta_{test} \subset \Theta$. The dynamics of a KG $\mathcal{G}$ are then simulated via the temporal transition from $\mathcal{G}_{t-1}$ with an embedding $\mathcal{E}_{t-1}$ at time $t-1$ towards $\mathcal{G}_t$ at time t by removing the test nodes in $\mathcal{G}_{t-1}$. The reconstructed embeddings of $\mathcal{G}_t$ based on $\mathcal{G}_{t-1}$ and $\mathcal{E}_{t-1}$ are denoted as $\Gamma_t^{(t-1)} = \{\gamma_n^{(t-1)}(v) : v \in V_t\}$. Subsequently, three different performances p are determined as accuracies of successive classifiers.

1. p_{sota}: The state of the art (sota) approach for integrating the new nodes Θ_{test} is to retrain an embedding $\mathcal{E}_t$ for the KG $\mathcal{G}_t$. The classifier is retrained on these embeddings $\{\epsilon_t(v) : v \in \Theta_{train}\}$ and evaluated for $\{\epsilon_t(v) : v \in \Theta_{test}\}$.
2. $p^{\text{add}}_{\text{navi}}$: A Navi Approach $\gamma_n^{(t-1)}$ is trained on $\mathcal{E}_{t-1}$ and $\mathcal{G}_{t-1}$. The classifier is trained on $\{\epsilon_{t-1}(v) : v \in \Theta_{train}\}$ and evaluated for $\{\gamma_n^{(t-1)}(v) : v \in \Theta_{test}\}$.
3. $p^{\text{full}}_{\text{navi}}$: The same Navi Approach $\gamma_n^{(t-1)}$ is assumed as for $p^{\text{add}}_{\text{navi}}$. However, the classifier is trained on the reconstructions $\{\gamma_n^{(t-1)}(v) : v \in \Theta_{train}\}$ as well.

For the evaluation, it is of relevance whether the performance $p^{\text{add}}_{\text{navi}}$ maintains a similar level as p_{sota} despite reusing the classifier and not retraining the static embedding model. This comparison is also needed for link predictions, although other metrics as performances must be considered in this case [19]. On the other hand, $p^{\text{full}}_{\text{navi}}$ evaluates a novel and dynamic embedding method by reconstructing the full graph and training a new classifier based on these reconstructions.

6 Preliminary Results

The preliminary results are initially limited to the AIFB KG and the entity classification task for a Support Vector Machine with RBF kernel and regularization $C = 1$. As static embedding methods, RDF2Vec [11], TransE [3], TransD [10], CrossE [24], RotatE [16], ConvE [4], RESCAL [8] and DistMult [23] are considered. The RDF2Vec embeddings were generated with pyrdf2vec [21] and the remaining embeddings via pykeen [1] with the default settings. Navi Reconstructions of these embeddings are always achieved via $\gamma(\cdot) := \gamma^{(wm)}\left(\Phi_C(\cdot), \Phi_T(\cdot)\right)$ with the Navi Merger $\gamma^{(wm)}$ and the Navi Layers Φ_C, Φ_T from Section 4. The training runs are performed using Adam, a learning rate of 0.001, dropout 0.5, the epochs $\{0, 100, 500\}$ and the following modified mean squared error (MSE)

$$\mathcal{L}\big(\gamma(x), \epsilon(x)\big) = \tfrac{1}{d}\sum_{j=1}^{d}\left\{\left|\tfrac{\exp(\min(\gamma(x)_j, \epsilon(x)_j))}{\exp(\max(\gamma(x)_j, \epsilon(x)_j))} - 1\right| + \big[\gamma(x)_j - \epsilon(x)_j\big]^2\right\}.$$

The exponential summand takes into account that, due to the squaring, absolutely small entries are mostly neglected by the standard MSE.

Table 1. Evaluation of the entity classification task. The 3-tuples $(p_{\text{sota}}, p^{\text{add}}_{\text{navi}}, p^{\text{full}}_{\text{navi}})$ represent the evaluation performances (here classification accuracies in percent). For RESCAL with the epoch number 1000 the performances $(53, 53, 67)$ were obtained.

Epochs	RDF2Vec	TransE	TransD	CrossE	RotatE	ConvE	RESCAL	DistMult
0	83, 89, 92	83, 72, 89	83, 86, 92	69, 44, 86	89, 56, 89	78, 42, 89	53, 31, 58	92, 47, 89
100	83, 86, 92	83, 89, 89	83, 86, 86	69, 53, 86	89, 86, 89	78, 83, 89	53, 33, 58	92, 83, 86
500	83, 89, 92	83, 89, 89	83, 86, 92	69, 81, 86	89, 89, 89	78, 86, 89	53, 47, 67	92, 92, 92

The results from Table 1 suggest that, at least in the present setting, the research questions RQ1 and RQ2 could be solved by the Navi Approach. Sufficiently large epoch numbers apparently lead to $p^{\text{add}}_{\text{navi}} \geq p_{\text{sota}}$, which exceeds the minimum goal of approximately maintaining the state of the art performance p_{sota} without the need of relearning the embedding model after each update to the graph. Further, the performances $p^{\text{full}}_{\text{navi}}$ suggest some sort of regularization of the data. Without a single training run, the use of Navi Reconstructions mostly yields better performances and simultaneously allows the desired dynamics in the KG. The still ongoing evaluation of the Navi Reconstructions implies that these results and observations also apply to the knowledge graphs MUTAG, BGS and AM.

7 Conclusions and Lessons Learned

In this paper, we presented the Navi Approach as a possible enabler of simplified and dynamic knowledge graph embeddings. It is gratifying and not to be taken for granted that these promising results are already available at such an early stage of this PhD, justifying further research in this direction. Thus, an extension of the evaluation to large-scale graphs like DBpedia, Wikidata and YAGO is planned already. In the process, further forms of dynamics are to be highlighted and tested. Based on this, we will investigate whether and to what extent the Navi Approach already provides an answer to the research questions RQ1 and RQ2. This phase of research will also clarify whether, as conjectured in Section 6, a form of regularization actually occurs when existing embeddings are replaced by their reconstructions. Furthermore, there is the possibility to use the Navi Approach as a simplified surrogate model of static embedding methods and thus try to improve the interpretability of KG embeddings in future work as well.
In summary, this PhD's research will contribute to the ability to process dynamic knowledge graphs, making them accessible to humans as well as machines as a basis for their communication. This result would represent a fundamental step towards current research fields like Industry 5.0 and at the same time have a major impact on all areas involving dynamic knowledge graphs.

Acknowledgements. This PhD is part of the TEAMING.AI project which receives funding in the European Commission's Horizon 2020 Research Programme under Grant Agreement Number 957402 (www.teamingai-project.eu). Furthermore, I would like to thank my supervisor Prof. Dr. Heiko Paulheim and my co-supervisor Dr. Tobias Weller for their continuous support.

References

1. Ali, M., Berrendorf, M., et al.: Pykeen: A Python Library for Training and Evaluating Knowledge Graph Embeddings. Journal of Machine Learning Research **22** (2021)
2. Bader, S.R., Grangel-Gonzalez, I., et al.: A knowledge graph for industry 4.0. In: The Semantic Web (2020)
3. Bordes, A., Usunier, N., et al.: Translating embeddings for modeling multi-relational data. In: Advances in Neural Information Processing Systems. vol. 26. Curran Associates, Inc. (2013)
4. Dettmers, T., Pasquale, M., et al.: Convolutional 2d knowledge graph embeddings. In: Proceedings of the 32th AAAI Conference on Artificial Intelligence (2018)
5. Diefenbach, D., Giménez-García, J., et al.: Qanswer kg: Designing a portable question answering system over rdf data. In: The Semantic Web. Springer (2020)
6. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015)
7. Nahavandi, S.: Industry 5.0 - a human-centric solution. Sustainability **11**(16) (2019)

8. Nickel, M., Tresp, V., et al.: A three-way model for collective learning on multi-relational data. In: Proceedings of the 28th International Conference on International Conference on Machine Learning. Omnipress (2011)
9. Palumbo, E., Rizzo, G., et al.: Knowledge graph embeddings with node2vec for item recommendation. In: The Semantic Web: ESWC 2018 Satellite Events. Springer International Publishing (2018)
10. Rahman, M.M., Takasu, A.: Knowledge graph embedding via entities' type mapping matrix. In: International Conference on Neural Information Processing. Springer (2018)
11. Ristoski, P., Paulheim, H.: Rdf2vec: Rdf graph embeddings for data mining. In: The Semantic Web – ISWC 2016. Springer International Publishing (2016)
12. Ristoski, P., de Vries, G.K.D., et al.: A collection of benchmark datasets for systematic evaluations of machine learning on the semantic web. In: The Semantic Web – ISWC 2016. Springer International Publishing (2016)
13. Sack, H., Biswas, R., Gesese, G.A., Alam, M.: A survey on knowledge graph embeddings with literals: Which model links better literal-ly? Semantic Web Journal (2020)
14. Sankar, A., Wu, Y., et al.: Dysat: Deep neural representation learning on dynamic graphs via self-attention networks. In: Proceedings of the 13th International Conference on Web Search and Data Mining (2020)
15. Schlichtkrull, M., Kipf, T.N., et al.: Modeling relational data with graph convolutional networks. In: The Semantic Web. Springer International Publishing (2018)
16. Sun, Z., Deng, Z., et al.: Rotate: Knowledge graph embedding by relational rotation in complex space. In: International Conference on Learning Representations (2019)
17. Tay, Y., Luu, A., et al.: Non-parametric estimation of multiple embeddings for link prediction on dynamic knowledge graphs. Proceedings of the AAAI Conference on Artificial Intelligence **31**(1) (2017)
18. Trivedi, R., Farajtabar, M., et al.: Dyrep: Learning representations over dynamic graphs. In: International conference on learning representations (2019)
19. Trouillon, T., Welbl, J., et al.: Complex embeddings for simple link prediction. In: Proceedings of the 33rd International Conference on International Conference on Machine Learning (2016)
20. Vaigh, C.B.E., Goasdoué, F., et al.: A novel path-based entity relatedness measure for efficient collective entity linking. In: International Semantic Web Conference. Springer (2020)
21. Vandewiele, G., Steenwinckel, B., et al.: pyRDF2Vec: Python Implementation and Extension of RDF2Vec (IDLab) (2020)
22. Wu, T., Khan, A., et al.: Efficiently embedding dynamic knowledge graphs. arXiv preprint arXiv:1910.06708 (2019)
23. Yang, B., Yih, W., et al.: Embedding entities and relations for learning and inference in knowledge bases. In: 3rd International Conference on Learning Representations (2015)
24. Zhang, W., Paudel, B., et al.: Interaction embeddings for prediction and explanation in knowledge graphs. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining. WSDM '19, Association for Computing Machinery (2019)