Context-Dependent Knowledge Graph Embedding

Yuanfei Luo[1,2], **Quan Wang**[1*], **Bin Wang**[1], **Li Guo**[1]
[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{luoyuanfei,wangquan,wangbin,guoli}@iie.ac.cn
[2]University of Chinese Academy of Sciences, Beijing, China

Abstract

We consider the problem of embedding knowledge graphs (KGs) into continuous vector spaces. Existing methods can only deal with explicit relationships within each triple, i.e., local connectivity patterns, but cannot handle implicit relationships across different triples, i.e., contextual connectivity patterns. This paper proposes context-dependent KG embedding, a two-stage scheme that takes into account both types of connectivity patterns and obtains more accurate embeddings. We evaluate our approach on the tasks of link prediction and triple classification, and achieve significant and consistent improvements over state-of-the-art methods.

1 Introduction

Knowledge Graphs (KGs) like WordNet (Miller, 1995), Freebase (Bollacker et al., 2008), and DBpedia (Lehmann et al., 2014) have become extremely useful resources for many NLP-related applications. A KG is a directed graph whose nodes correspond to entities and edges to relations. Each edge is a triple of the form (h, r, t), indicating that entities h and t are connected by relation r. Although powerful in representing complex data, the symbolic nature makes KGs hard to manipulate.

Recently, knowledge graph embedding has attracted much attention (Bordes et al., 2011; Bordes et al., 2013; Socher et al., 2013; Wang et al., 2015). It attempts to embed entities and relations in a KG into a continuous vector space, so as to simplify the manipulation while preserving the inherent structure of the original graph.

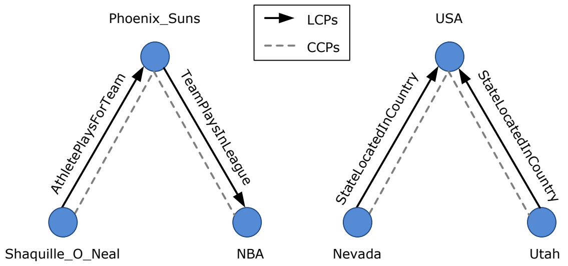


Figure 1: LCPs and CCPs.

Most of the existing KG embedding methods model triples individually, ignoring the fact that entities connected to a same node are usually implicitly related to each other, even if they are not directly connected. Figure 1 gives two examples. `Shaquille_O_Neal` and `NBA` in the former example and `Nevada` and `Utah` in the latter example are implicitly related to each other, through the intermediate nodes `Phoenix_Suns` and `USA` respectively. We refer to such implicit relationships as *contextual connectivity patterns* (CCPs). Relationships explicitly represented in triples are referred to as *local connectivity patterns* (LCPs). In most of the existing methods, only LCPs are explicitly modeled.

This paper proposes a two-stage embedding scheme that explicitly takes into account both CCPs and LCPs, called *context-dependent KG embedding*. In the first stage, each CCP is formalized as a *knowledge path*, i.e., a sequence of entities and relations occurring in the pattern. A word embedding model is adopted to learn embeddings of entities and relations, by taking them as pseudo-words. The embeddings are enforced compatible within each knowledge path, and hence can capture CCPs. In the second stage, the learned embeddings are fine-tuned by an existing KG embedding technique. Since such a technique requires the embeddings to be compatible on each individual triple, LCPs are also encoded.

The advantages of our approach are three-fold. 1) It fully exploits both CCPs and LCPs, and can

[*]Corresponding author: Quan Wang.

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 1679–1684,
Lisbon, Portugal, 17-21 September 2015. ©2015 Association for Computational Linguistics.

obtain more accurate embeddings. 2) It is a general scheme, applicable to a wide variety of word embedding models in the first stage and KG embedding models in the second. 3) No auxiliary data is further required in the two-stage process, except for the original graph.

We evaluate our approach on two publicly available data sets, and achieve significant and consistent improvements over state-of-the-art methods in the link prediction and triple classification tasks. The learned embeddings are not only more accurate but also more stable.

2 Context-Dependent KG Embedding

We are given a KG with nodes corresponding to entities and edges to relations. Each edge is denoted by a triple (h, r, t), where h is the head entity, t the tail entity, and r the relation between them. Entities and relations are represented as vectors, matrices, or tensors in a continuous vector space. Context-dependent KG embedding aims to automatically learn entity and relation embeddings, by using observed triples $\mathcal{O}$ in a two-stage process.

2.1 Modeling CCPs

The first stage models CCPs conveyed in the KG. Each CCP is formalized as a knowledge path, i.e., a sequence of entities and relations occurring in the pattern. For the CCPs in Figure 1, the associated knowledge paths are:

“Shaquille_O_Neal, AthletePlaysForTeam, Phoenix_Suns, TeamPlaysInLeague, NBA”

“Nevada, StateLocatedInCountry, USA, StateLocatedInCountry, Utah”.

We fix the length of knowledge paths to 5. During path extraction, we ignore the directionality of edges, and treat the KG as an undirected graph.[1]

Given the extracted knowledge paths, we employ word embedding models to pre-train the embeddings of entities and relations, by taking them as pseudo-words. We use two word embedding models: CBOW and Skip-gram (Mikolov et al., 2013a; Mikolov et al., 2013b). In CBOW, words in the context are projected to their embeddings and then summed. Based on the summed embedding, log-linear classifiers are employed to predict the current word. In Skip-gram, the current word is projected to its embedding, and log-linear classifiers are further adopted to predict its context. We restrain the context of a word (i.e. entity/relation) within each knowledge path. The entity and relation embeddings pre-trained in this way are required to be compatible within each knowledge path, and thus can encode CCPs.

Perozzi et al. (2014) and Goikoetxea et al. (2015) have proposed similar ideas, i.e., to generate random walks from online social networks or from the WordNet knowledge base, and then employ word embedding techniques on these random walks. But our approach has two differences. 1) It deals with heterogeneous graphs with different types of edges. Both nodes (entities) and edges (relations) are included during knowledge path extraction. However, the previous studies focus only on nodes. 2) We devise a two-stage scheme where the embeddings learned in the first stage will be fine-tuned in the second one, while the previous studies take such embeddings as final output.

2.2 Modeling LCPs

The second stage models LCPs conveyed in the KG. We employ three state-of-the-art KG embedding models, namely SME (Bordes et al., 2014), TransE (Bordes et al., 2013), and SE (Bordes et al., 2011) to fine-tune the pre-trained embeddings. These three models work in the following way. First, entities are represented as vectors, and relations as operators in an embedding space, characterized by vectors (SME and TransE) or matrices (SE). Then, for each triple (h, r, t), an energy function $f_r(h, t)$ is defined to measure its plausibility. Plausible triples are assumed to have low energies. Finally, to obtain entity and relation embeddings, a margin-based ranking loss, i.e.,

$$\mathcal{L} = \sum_{t^+ \in \mathcal{O}} \sum_{t^- \in \mathcal{N}_{t^+}} \left[\gamma + f_r(h, t) - f_r(h', t')\right]_+,$$

is minimized. Here, $t^+ = (h, r, t) \in \mathcal{O}$ is an observed (positive) triple; $\mathcal{N}_{t^+}$ is the set of negative triples constructed by replacing entities in t^+, and $t^- = (h', r, t') \in \mathcal{N}_{t^+}$; γ is a margin separating positive and negative triples; $[x]_+ = \max(0, x)$. Table 1 summarizes the entity/relation embeddings and the energy functions used in SME, TansE, and SE. For other KG embedding models, please refer to (Nickel et al., 2011; Riedel et al., 2013; Wang et al., 2014; Chang et al., 2014).

We adopt stochastic gradient descent to solve the minimization problem, by taking entity and relation embeddings pre-trained in the first stage as

[1] Two entities connected to a same node are always expected to have some implicit relationships, no matter how they are connected to the intermediate node.

Method	Entity/Relation embedding	Energy function
SME (linear) (Bordes et al., 2014)	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^k, \mathbf{r} \in \mathbb{R}^k$	$f_r(\mathbf{h}, \mathbf{t}) = (\mathbf{W}_{u1}\mathbf{r} + \mathbf{W}_{u2}\mathbf{h} + \mathbf{b}_u)^T (\mathbf{W}_{v1}\mathbf{r} + \mathbf{W}_{v2}\mathbf{t} + \mathbf{b}_v)$
SME (bilinear) (Bordes et al., 2014)	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^k, \mathbf{r} \in \mathbb{R}^k$	$f_r(\mathbf{h}, \mathbf{t}) = ((\underline{\mathbf{W}}_u \bar{\times}_3 \mathbf{r})\mathbf{h} + \mathbf{b}_u)^T ((\underline{\mathbf{W}}_v \bar{\times}_3 \mathbf{r})\mathbf{t} + \mathbf{b}_v)$
TransE (Bordes et al., 2013)	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^k, \mathbf{r} \in \mathbb{R}^k$	$f_r(\mathbf{h}, \mathbf{t}) = \Vert\mathbf{h} + \mathbf{r} - \mathbf{t}\Vert_{\ell_1}$
SE (Bordes et al., 2011)	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^k, \mathbf{R}_u, \mathbf{R}_v \in \mathbb{R}^{k \times k}$	$f_r(\mathbf{h}, \mathbf{t}) = \Vert\mathbf{R}_u\mathbf{h} - \mathbf{R}_v\mathbf{t}\Vert_{\ell_1}$

Table 1: Entity/Relation embeddings and energy functions used in KG embedding methods.

	# rel.	# ent.	# trip. (train/valid/test)			# path
WN18	18	40,943	141,442	5,000	5,000	5,674,308
NELL186	186	14,463	31,134	5,000	5,000	1,914,475

Table 2: Statistics of the data sets.

initial values.[2] The entity and relation embeddings fine-tuned in this way are required to be compatible within each triple, and thus can encode LCPs.

Socher et al. (2013) have proposed a similar idea, i.e., to use embeddings learned from an auxiliary corpus as initial values. However, linking entities recognized in an auxiliary corpus to those occurring in the KG is always a non-trivial task. Our approach requires no auxiliary data, and naturally avoids the entity linking task.

3 Experiments

We test our approach on the tasks of link prediction and triple classification. Two publicly available data sets are used. The first is WN18 released by Bordes et al. (2013)[3]. It is a subset of WordNet, consisting of 18 relations and the entities connected by them. The second is NELL186 released by Guo et al. (2015)[4], containing the most frequent 186 relations in NELL (Carlson et al., 2010) and the associated entities. Triples are split into training/validation/test sets, used for model training, parameter tuning, and evaluation respectively. Knowledge paths are extracted from training sets. Table 2 gives some statistics of the data sets.

To perform context-dependent KG embedding, we use CBOW and Skip-gram in the pre-training stage, and SME, TransE, and SE in the fine-tuning stage. We take randomly initialized SME, TransE, and SE as baselines, denoted as *-Random. We do not compare to the setting that employs only CBOW or Skip-gram, since it does not provide an energy function to calculate triple plausibility, which hinders the evaluation of both tasks.

3.1 Link Prediction

Link prediction is to predict whether there is a specific relation between two entities.

Evaluation Protocol. For each test triple, the head is replaced by every entity in the KG, and the energy is calculated for each corrupted triple. Ranking the energies in ascending order, we get the rank of the correct answer. We can get another rank by corrupting the tail. We report two metrics on the test sets: Mean (averaged rank) and Hits@10 (proportion of ranks no larger than 10).

Implementation Details. To train CBOW and Skip-gram, we use the word2vec implementations[5]. 20 negative samples are drawn for each positive one. The context size is fixed to 5. To train SME, TransE, and SE, we use the implementations provided by the authors[6], with 100 mini-batches. We vary the learning rate in $\{0.01, 0.1, 1, 10\}$, the dimension k in $\{20, 50\}$, and the margin γ in $\{1, 2, 4\}$. The best model is selected by monitoring Hits@10 on the validation sets, with a total of at most 1000 iterations over the training sets.

Results. Table 3 reports the results on the test sets of WN18 and NELL186. The improvements of CBOW/Skip-gram over Random are also given. Statistically significant improvements are marked by ‡ (sign test, significance level 0.05). The results show that a pre-training stage consistently improves over the baselines for all the methods on both data sets. Almost all of the improvements are statistically significant.

3.2 Triple Classification

Triple classification aims to verify whether an unseen triple is correct or not.

Evaluation Protocol. Triples in the validation and test sets are labeled as positive instances. For each positive instance, we construct a negative instance by randomly corrupting the entities. During

[2] For SE, only entity vectors are initialized by pre-trained embeddings. Relation matrices are randomly initialized.

[3] https://everest.hds.utc.fr/doku.php?id=en:smemlj12

[4] http://www.aclweb.org/anthology/P/P15/

[5] https://code.google.com/p/word2vec/

[6] https://github.com/glorotxa/SME

		Mean			Hits@10 (%)		
		Random	CBOW	Skip-gram	Random	CBOW	Skip-gram
WN18	SME (linear)	463.2	‡286.5 (↓38%)	226.9 (↓51%)	63.98	‡68.65 (↑7%)	‡70.01 (↑9%)
	SME (bilinear)	551.8	‡308.8 (↓44%)	‡279.2 (↓49%)	63.83	‡67.65 (↑6%)	‡67.53 (↑6%)
	TransE	723.1	‡293.0 (↓59%)	‡290.0 (↓60%)	78.50	‡79.67 (↑1%)	‡79.87 (↑2%)
	SE	960.0	‡426.2 (↓56%)	‡289.4 (↓70%)	71.53	‡76.05 (↑6%)	‡75.89 (↑6%)
NELL186	SME (linear)	595.5	‡371.9 (↓38%)	‡340.3 (↓43%)	29.82	‡34.22 (↑15%)	‡35.57 (↑19%)
	SME (bilinear)	375.2	‡305.0 (↓19%)	‡292.9 (↓22%)	37.45	‡39.31 (↑ 5%)	‡39.70 (↑ 6%)
	TransE	732.6	‡384.6 (↓48%)	‡384.6 (↓48%)	27.60	‡28.71 (↑ 4%)	‡30.52 (↑11%)
	SE	2307.0	‡1314.7 (↓43%)	‡412.2 (↓82%)	19.53	‡26.15 (↑34%)	‡31.12 (↑59%)

Table 3: Link prediction results on the test sets of WN18 and NELL186.

		Micro-ACC (%)			Macro-ACC (%)		
		Random	CBOW	Skip-gram	Random	CBOW	Skip-gram
WN18	SME (linear)	84.70	89.54 (↑6%)	89.16 (↑5%)	85.11	89.11 (↑5%)	90.57 (↑6%)
	SME (bilinear)	84.30	91.83 (↑9%)	90.68 (↑8%)	85.36	90.49 (↑6%)	89.89 (↑5%)
	TransE	94.60	96.98 (↑3%)	97.23 (↑3%)	86.74	93.46 (↑8%)	94.49 (↑9%)
	SE	94.71	96.46 (↑2%)	96.42 (↑2%)	87.99	92.05 (↑5%)	91.70 (↑4%)
NELL186	SME (linear)	88.59	89.95 (↑2%)	91.19 (↑3%)	84.42	85.70 (↑2%)	86.67 (↑3%)
	SME (bilinear)	88.74	93.22 (↑5%)	92.86 (↑5%)	83.41	89.70 (↑8%)	89.65 (↑7%)
	TransE	82.54	85.65 (↑4%)	85.33 (↑3%)	76.74	80.06 (↑4%)	80.06 (↑4%)
	SE	89.00	93.37 (↑5%)	93.07 (↑5%)	83.01	87.89 (↑6%)	87.98 (↑6%)

Table 4: Triple classification results on the test sets of WN18 and NELL186.

classification, a triple is predicted to be positive if the energy is below a relation-specific threshold δ_r; otherwise negative. We report two metrics on the test sets: micro-averaged accuracy (per-instance average) and macro-averaged accuracy (per-relation average).

Implementation Details. We use the same parameter settings as in the link prediction task. The relation-specific threshold δ_r is determined by maximizing Micro-ACC on the validation sets.

Results. Table 4 reports the results on the test sets of WN18 and NELL186. The results again demonstrate both the superiority and the generality of our approach.

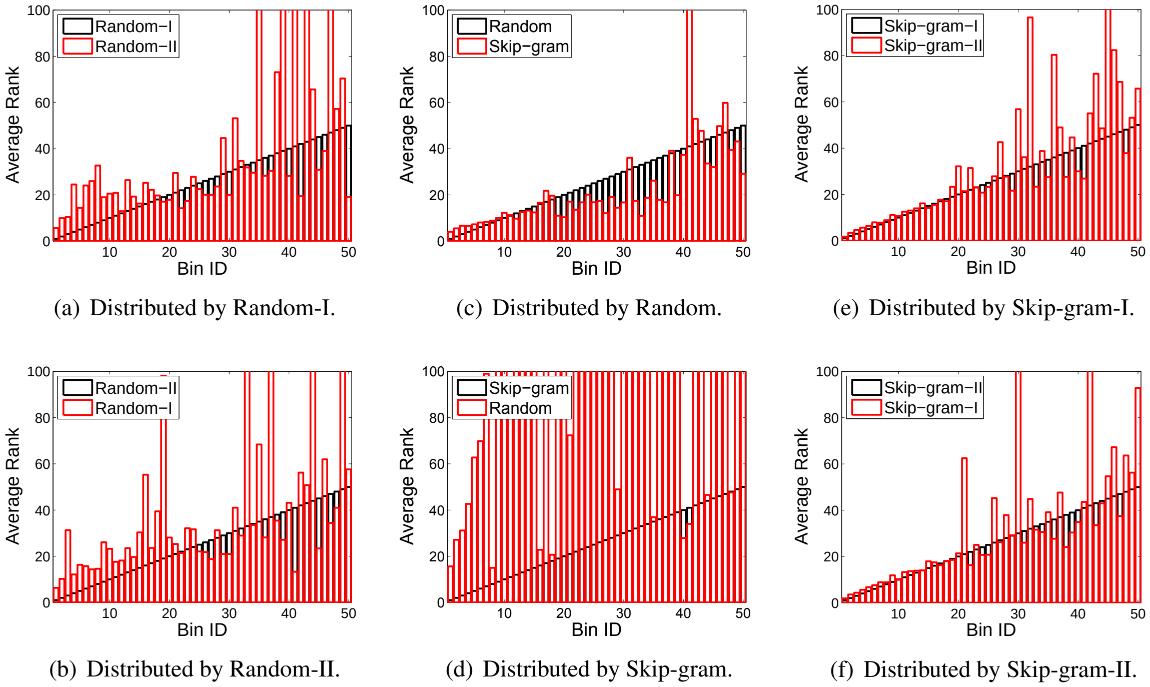
3.3 Discussions

This section is to explore why pre-training helps in KG embedding, specifically in link prediction.

We first test different random initializations in traditional KG embedding models. We run SME (linear) twice on WN18, with two different initialization settings. Both are randomly sampled from the same uniform distribution, but with different seeds, referred to as Random-I and Random-II. Each setting finally gets 10,000 ranks on the test set.[7] To better understand the difference between the two settings, we analyze the ranks individually, rather than reporting aggregated metrics (Mean and Hits@10). Specifically, we distribute the 10,000 instances into different bins according to the ranks given by one setting (e.g. Random-I). Instances assigned to the i-th bin have the same rank of i, that means, they are all ranked in the i-th position by this setting. Then, within each bin, we calculate the average rank of the instances given by the other setting (e.g. Random-II). If the average rank differs drastically from the bin ID, the instances in this bin are ranked significantly differently by the two settings. Figures 2(a) and 2(b) show the results, with the instances distributed according to Random-I and Random-II respectively. In both cases, we retain the bins with ID no larger than 50, covering about 85% of the instances. In most of the bins, the average rank (red bars in the figures) differs drastically from the bin ID (black bars in the figures), indicating that the ranks given by Random-I and Random-II are significantly different at the instance level. The results demonstrate the non-convexity of SME (linear): different initial values lead to different local minimum.

We further compare the settings of initial values 1) randomly sampled from a uniform distribution (Random) and 2) pre-trained by Skip-gram

[7] For each of the 5,000 test triples, both the head and the tail are corrupted and ranked.



(a) Distributed by Random-I.

(c) Distributed by Random.

(e) Distributed by Skip-gram-I.

(b) Distributed by Random-II.

(d) Distributed by Skip-gram.

(f) Distributed by Skip-gram-II.

Figure 2: Ranks obtained by different initialization strategies (best viewed in color).

(Skip-gram). The results are given in Figures 2(c) and 2(d). In most of the bins Skip-gram has an average rank lower than the bin ID (Figure 2(c)), while Random has an average rank much higher than the bin ID (Figure 2(d)), implying that Skip-gram performs better than Random-I at the instance level. The results indicate that pre-training might help in finding better initial values which lead to better local minimum.

Finally we test our two-stage KG embedding scheme where the skip-gram model itself is given two different initialization settings, say Skip-gram-I and Skip-gram-II. The results are given in Figures 2(e) and 2(f). In each of the first 20 bins, Skip-gram-I and Skip-gram-II get an average rank almost the same with the bin ID, implying that the two settings perform quite similarly, particularly at the highest ranking levels. The results indicate that a pre-training stage might help in obtaining more stable embeddings.

4 Conclusion

We have proposed a novel two-stage scheme for KG embedding, called context-dependent KG embedding. In the pre-training stage CCPs are encoded by a word embedding model, and in the fine-tuning stage LCPs are encoded by a traditional KG embedding model. Since both types of connectivity patterns are explicitly taken into account, our approach can obtain more accurate embeddings. Moreover, our approach is quite general, applicable to various word embedding and KG embedding models. Experimental results on link prediction and triple classification demonstrate the superiority, generality, and stability of our approach.

As future work, we plan to 1) Investigate the efficacy of longer CCPs (i.e. knowledge paths with lengths longer than 5). 2) Design a joint model that encodes LCPs and CCPs simultaneously. Moreover, our approach actually reveals the possibility of a broad idea, i.e., initializing an embedding model by another embedding model. We would also like to test the feasibility of other such strategies, e.g., initializing SME by TransE, so as to combine the benefits of both models.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments and suggestions. This work is supported by the National Natural Science Foundation of China (grant No. 61402465), the Strategic Priority Research Program of the Chinese Academy of Sciences (grant No. XDA06030200), and the National Key Technology R&D Program (grant No. 2012BAH46B03).

References

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim S-turge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pages 1247–1250.

Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. 2011. Learning structured embeddings of knowledge bases. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence*, pages 301–306.

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*, pages 2787–2795.

Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. 2014. A semantic matching energy function for learning with multi-relational data. *Machine Learning*, 94(2):233–259.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Bur-r Settles, Estevam R. Hruschka Jr., and Tom M. Mitchell. 2010. Toward an architecture for never-ending language learning. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pages 1306–1313.

Kai-Wei Chang, Wen-tau Yih, Bishan Yang, and Christopher Meek. 2014. Typed tensor decomposition of knowledge bases for relation extraction. In *Proceedings of the 2014 Conference on Empirical Methods on Natural Language Processing*, pages 1568–1579.

Josu Goikoetxea, Aitor Soroa, and Eneko Agirre. 2015. Random walks and neural network language models on knowledge bases. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1434–1439.

Shu Guo, Quan Wang, Bin Wang, Lihong Wang, and Li Guo. 2015. Semantically smooth knowledge graph embedding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 84–94.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, et al. 2014. Dbpedia: A large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at International Conference on Learning Representations*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

George A Miller. 1995. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–41.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on Machine Learning*, pages 809–816.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKD-D International Conference on Knowledge Discovery and Data Mining*, pages 701–710.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference on North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 74–84.

Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems*, pages 926–934.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pages 1112–1119.

Quan Wang, Bin Wang, and Li Guo. 2015. Knowledge base completion using embeddings and rules. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pages 1859–1865.