Ensemble Solutions for Link-Prediction in Knowledge Graphs

Denis Krompaß[1,2] and Volker Tresp[1,2]

[1] Ludwig Maximilian University, 80538 Munich, Germany
[2] Corporate Technology, Siemens AG, Munich, Germany

Abstract. Knowledge graphs store up to billions of facts and are used in an increasing number of applications including search and question answering. Besides the freely available knowledge graphs of the Linked Open Data Cloud, there are a number of commercial knowledge graphs as the Google Knowledge Graph and Microsoft's Satori. Incompleteness is still a major issue in this graphs and has been the topic of a multitude of research efforts in the recent years; latent variable models were shown to be very effective by complementing knowledge graphs via link-prediction. Prior works on link-prediction with latent variable models mostly focused on proposing new variants or extensions, but disregarded the potential of combining various approaches to drive prediction quality. In this work we study the potential of combining existing state of the art latent variable models for link-prediction. We show on various datasets extracted from DBpedia, YAGO and Freebase that these models indeed complement each other, leading to large improvements of the best single predictor of up to 11%.

1 Introduction

In the recent years, it was shown that domains such as natural language processing, search or question answering all benefit to a large extend from information provided by knowledge graphs. IBM's Watson system or Google's web-search are two of the most prominent examples where the integration of knowledge-graphs has been proven to be of great success.

Today's knowledge-graphs depend on a large community of voluntary contributors that directly add or edit facts in the knowledge-graph or its sources. Freebase [2] for example allows users to directly add and edit facts in Freebase, but it also automatically extracts facts from other hand-curated sources like Wikipedia Infoboxes or MusicBrainz. Thanks to the great efforts of these contributors, some knowledge graphs have already reached an impressive size containing billions of facts that express thousands of different relations between millions of entities. Nevertheless, the included knowledge in these databases suffers from incompleteness and can be outdated or faulty, especially in the cases of less popular entities. As an example, a vast amount of person entities in Freebase (71% [20]) or DBpedia (66% [8]) are missing a place of birth.

Latent variable models (LVM) are a class of machine-learning algorithms which have been successfully exploited in knowledge-graphs. These models show an especially high potential for link-prediction, which can be exploited for cleaning and completion of these databases; another application is the support for an automatic knowledge graph construction system from unstructured text data [20]. In [10] it was shown that these models can be used for querying probabilistic knowledge graphs.

From our perspective, the current state of the art for link-prediction in web-scale knowledge-graphs with LVMs is represented by RESCAL [14], TransE [3] and the multiway neural network proposed in [20]. In this paper we argue that these models learn different patterns in the data that complement each other. We study these complementary effects for link-prediction by constructing and analyzing simple ensembles of these different methods. In addition we exploit domain and range constraints in these models in form of type-constraints extracted from the knowledge graph's schema or approximated through a local closed-world assumption. It was shown in [8] that prior knowledge about these constraints significantly drives the prediction quality of LVMs. We evaluate the ensembles on various datasets extracted from popular knowledge-bases including Freebase [2], DBpedia [1] and YAGO [7].

The paper is structured as follows: In the next section we will provide related work. In Section 2 we will briefly review RESCAL, Translational Embeddings and the multiway neural network approach exploited in the Knowledge Vault project and motivate and describe the ensemble of these methods in Section 3. Subsequent to that we will provide and discuss our experimental results in Section 4. We conclude in Section 5.

1.1 RelatedWork

There is a large body of work on link prediction in knowledge-bases (we refer to [13] for a review). A neural tensor network for link prediction in knowledge bases was proposed by [18]. The performance of this model could be reproduced by a simpler neural network model by [20] in the Google Knowledge Vault project. Very recently, [21] proposed a framework for relationship modeling that combines aspects of the link-prediction models introduced in [3] and [18]. [19, 12] proposed TransH and TransR which improves TransE's capability to model reflexive one-to-many, many-to-one and many-to-many relation-types. In [5] a combination of a trigram and bigram model for link-prediction is proposed (TATEC), where both models are trained independently in a pre-training phase before they are jointly optimized. [6] used a bagging approach to combine various classification algorithms (e.g. Support-Vector-Machines, K-Nearest-Neighbor) for link-prediction, but could not observe any significant improvements over the best single predictors. The integration of prior knowledge about relation-types into RESCAL [4, 9] and generally in latent variable models [8] was recently shown to significantly improve link-prediction quality of these models. General methods for link-prediction also include Markov-Logic-Networks [17] which have a limited scalability and random walk algorithms like the path ranking algorithm [11].

2 Latent Variable Models for Link Prediction

In the following we very briefly review the latent variable models targeted in this work. We refer to [8, 13, 3, 20, 14] for more details on these models.

2.1 RESCAL

RESCAL [14] is a three-way-tensor factorization model that has been shown to lead to very good results in various canonical relational learning tasks like link prediction, entity resolution and collective classification [15]. One main feature of RESCAL is that it can exploit a collective learning effect when used on relational data, since an entity has a unique representation over occurrences as a subject or as an object in a relationship and also over all relation types (properties) in the data. For RESCAL, triples are stored in an third-order adjacency tensor. RESCAL computes a d-rank factorization of this adjacency tensor by minimizing a regularized least-squares loss function through Alternating Least-Squares (ALS). For minimization, RESCAL uses closed world assumptions and therefore treats any unobserved triple as negative evidence. The confidence $\theta_{s,p,o}$ of RESCAL into a triple (s, p, o) is then given by

$$\theta_{s,p,o} = \mathbf{a}_s^T \mathbf{R}_p \mathbf{a}_o, \tag{1}$$

where $\mathbf{a}_s$ and $\mathbf{a}_o$ are the learned latent embedding vectors for the subject and object entities s and o, respectively, and $\mathbf{R}_p$ is a $d \times d$ matrix representing the latent embedding for the predicate relation-type p

2.2 Translational Embeddings Model

In [3] an energy-based model was proposed for learning low-dimensional embeddings of entities, where relationships are represented as translations in the embedding space (*TransE*). The approach assumes for a true fact that a property specific translation function exists that is able to map (or translate) the latent vector representation of the subject entity to the latent representation of the object entity. The confidence into a fact is expressed by the similarity of the translation of the subject embedding to the object embedding. In case of TransE, the translation function is defined by a simple addition of the latent vector representations of the subject entity $\mathbf{a}_s$ and predicate property $\mathbf{r}_p$. The similarity of the translation and the object embedding is measured by the $L1$ or $L2$ norm. TransE's confidence $\theta_{s,p,o}$ in a triple (s, p, o) is derived as

$$\theta_{s,p,o} = -\delta(\mathbf{a}_s + \mathbf{r}_p, \mathbf{a}_o), \tag{2}$$

where δ is the $L1$ or the $L2$ norm and $\mathbf{a}_o$ the latent embedding for the object entity. The embeddings are learned by minimizing a margin-based ranking cost function on a set of observed training triples T with stochastic gradient descent. Through the ranking cost function, unobserved triples are all treated as missing.

2.3 Knowledge Vault Multiway Neural Network

The Google Knowledge Vault project [20] pursues an automatic construction of a high quality knowledge graph. In this regard a neural network based model (denoted as multiway neural network *mwNN* in this work) for predicting prior probabilities for triples from existing knowledge graph data was proposed to support triple extraction from unstructured web documents. The confidence value $\theta_{s,p,o}$ for a target triple (s, p, o) is given by

$$\theta_{s,p,o} = \sigma(\beta^T \phi\left(\mathbf{W}\left[\mathbf{a}_s, \mathbf{r}_p, \mathbf{a}_o\right]\right)), \tag{3}$$

where $\phi()$ is the *tanh* function, $\mathbf{a}_s$ describes the latent embedding vector for the subject entity ($\mathbf{a}_o$ for the object entity) and $\mathbf{r}_p$ is the latent embedding for the predicate property p, which are stacked into a long column vector (denoted by $[\mathbf{a}_s, \mathbf{r}_p, \mathbf{a}_o]$). $\mathbf{W}$ and β are neural network weights and $\sigma()$ denotes the logistic function. The model is trained by minimizing a logistic cost-function through stochastic gradient descent. In difference to TransE, only object entities are corrupted and the corrupted triples are treated as negative evidence.

3 A Simple Ensemble for Link-Prediction

In this section we motivate an ensemble consisting of the three latent variable methods discussed in this work, RESCAL, TransE and mwNN. The main characteristic of a good ensemble is its composition out of very diverse single predictors that learn and recognize different patterns in the data. Through the diversity of the different predictors, complementary effects can be observed that drive overall prediction quality. In general, we also see a large diversity between RESCAL, TransE and mwNN. RESCAL assumes normally distributed variables minimizing a least-squares loss function, where mwNN assumes Bernoulli distributed variables minimizing a logistic loss function. TransE on the other hand minimizes a max-margin based ranking loss function. Further, RESCAL is a third-order tensor factorization method that is optimized through alternating least-squares with closed form solutions, where TransE is a distance based model and mwNN a neural network that are both optimized through stochastic gradient-descent. In addition, mwNN and TransE differ in the way they sample corrupted triples. In TransE, two corrupted triples are sampled for each true triple, where in each corrupted triple the subject or object entity is corrupted. mwNN generates negative triples by only corrupting the object entities through a randomly sampled entity. These methods also exploit different kinds of regularization; TransE projects the latent embeddings of entities on the $L2$ unit ball after each iteration during optimization and RESCAL and mwNN are typically minimized using $L1$ or $L2$ regularization on all parameters of the model.

For our study, we build simple ensembles in which we combine the link-predictions of RESCAL, TransE and mwNN. The final probability of a triple is then derived from the ensemble of these predictions by

$$P(x_{s,p,o} = 1|\Theta) = \frac{1}{n} \sum_{\theta^m \in \Theta}^{n} P(x_{s,p,o}|\theta^m_{s,p,o}) \tag{4}$$

$$\text{where} \qquad \Theta \subseteq \{\theta^{RESCAL}, \theta^{TransE}, \theta^{mwNN}\}$$

$$\text{and} \quad P(x_{s,p,o} = 1|\theta^m_{s,p,o}) = \frac{1}{1 + \exp\{-(\omega^m_1 \theta^m_{s,p,o} + \omega^m_0)\}} \tag{5}$$

$$\begin{aligned} \text{with} \qquad \theta^{RESCAL}_{s,p,o} &= \mathbf{a}_s^T \mathbf{R}_p \mathbf{a}_o \text{ (Equation 1)}, \\ \theta^{TransE}_{s,p,o} &= -\delta(\mathbf{a}_s + \mathbf{r}_p, \mathbf{a}_o) \text{ (Equation 2)}, \\ \theta^{mwNN}_{s,p,o} &= \sigma(\beta^T \phi\left(\mathbf{W}\left[\mathbf{a}_s, \mathbf{r}_p, \mathbf{a}_o\right]\right)) \text{ (Equation 3)}, \end{aligned}$$

where $x_{s,p,o}$ is the target variable that indicates if a triple (s, p, o), consisting of the subject and object entities s and o and the predicate relation-type p, is true.

Θ holds the pool of model parameters the ensemble combines for the prediction. We are evaluating all possible combinations of models, therefore Θ can be just a subset of the three, RESCAL, TransE and mwNN. For the ensemble, we fit and find the best hyper-parameters for the parameters of each model θ^{RESCAL}, θ^{TransE} and θ^{mwNN} independently, but the predicted confidence scores for triples generally differ between all model; mwNN predicts values between 0 and 1, where RESCAL can return any value in $\mathbb{R}$ and TransE returns negative distances. We could have applied a simple meta-learner, e.g. a simple logistic regression or a feed-forward neural network with one hidden layer to auto-balance the outputs of the tree methods, but we expected that such a meta learner could blur the individual contribution of each single-predictor in the link-prediction tasks. We used a Platt-Scaler [16] for each model based on a small subsample of the training data to get properly scaled probabilities (Equation 5). A Platt-Scaler is basically a logistic regression model that takes exactly one input (the output $\theta^m_{s,p,o}$ of the model m) and maps into the interval $[0, 1]$. The scalars ω^m_1 and ω^m_0 in Equation 5 denote the learned weight and bias of the logistic regression (Platt-Scaler) for the model m.

In difference to the other two methods, mwNN already predicts probabilities for triples. Nevertheless, we also learned a Platt-Scaler for this model in order to calibrate the probabilities of all models on the same dataset. For the final probability of a triple (s, p, o) we apply the scalers to the confidence score $\theta^m_{s,p,o}$ of each model m to get the probability $P(x_{s,p,o}|\theta^m_{s,p,o})$, that is the probability of the triple (s, p, o) given the model m. Subsequent to that, we simply combine each of these probabilities by computing the arithmetic mean (Equation 4).

4 Experiments

In our experiments we study empirically if TransE, RESCAL and mwNN are good targets for combination to drive link-prediction quality. We further study the value of each model for the ensemble by systematically discarding models

Table 1. Datasets used in the experiments.

Dataset	Entities	Properties	Triples
DBpedia-Music	321,950	15	981,383
Freebase-150k	151,146	285	1,047,844
YAGOc-195k	195,639	32	1,343,684

from the ensemble. Since it was shown by [8] that latent variable models benefit to a large extend from prior knowledge on domain and range constraints of relation-types, the ensembles are evaluated in two settings. In the first setting, the models exploit type-constraints extracted from the knowledge-base schema and in the second setting they use the local closed-world assumption (LCWA) [9, 8]. As discussed in [8], the LCWA has the advantage that it can be exploited without the need of prior knowledge about neither the types of the entities nor the typing of the properties. Therefore it can be applied to single properties or whole knowledge-graphs where information about type-constraints is absent or fuzzy.

4.1 Experimental Setup

In our experiments we used datasets extracted from Freebase [2], DBpedia [1] and YAGO [7] (Table 1). We refer to [8] for details on the extraction. Basically, the Freebase-150k datasets contains various different relation-types and entities and simulates a general purpose knowledge graph, where for DBpedia-Music only relation-types and entities that are related to music were extracted. The YAGOc-195k dataset contains triples from the high quality core dataset of YAGO.

We evaluate the different ensembles on link-prediction tasks using the same evaluation procedure as described in [8]; We split the triples into a holdout, validation and training set, where the first contains 20%, the second 10% and the latter 70% of the triples[3] and performed hyper-parameter using the train and validation set. After hyper-parameter tuning, we retrained all models using the best hyper-parameters on the combined datasets of validation and training set, thereby using 5% of triples (with additional negative sampling) for learning the Platt-Scalers for each model (RESCAL, TransE and mwNN). We report the Area Under Precision Recall Curve (AUPRC) score for each ensemble on the holdout set. In addition we report the AUPRC for the best single predictor as comparison.

4.2 Explanation of the Results Table Structure

In Table 2 the AUPRC results on the extracted datasets from Freebase, YAGO and DBpedia for the first setting are shown, where all models exploit the type-constraints given by the schema of the knowledge graph. Table 3 shows the same

[3] We additionally sampled 10 times as many negative triples for each set, where these sets of negative triples are not overlapping.

Table 2. AUPRC results on datasets, exploiting **Type-Constraints** in the models. **Model←*d*** indicates the dimensionality of the latent embeddings used by the models.

	Dataset		
Model←*d*=10	Freebase-150k	DBpedia-Music	YAGOc-195k
ALL	**0.846**	0.815	**0.883**
mwNN + TransE	0.820	**0.817**	0.881
mwNN + RESCAL	0.795	0.519	0.821
TransE + RESCAL	0.757	0.748	0.859
Best Single Predictor	(mwNN) 0.775	(TransE) 0.734	(TransE) 0.843
Model←*d*=50	Freebase-150k	DBpedia-Music	YAGOc-195k
ALL	**0.892**	**0.827**	**0.902**
mwNN + TransE	0.876	0.825	0.900
mwNN + RESCAL	0.835	0.756	0.845
TransE + RESCAL	0.819	0.783	0.891
Best Single Predictor	(mwNN) 0.815	(TransE) 0.783	(TransE) 0.896
Model←*d*=100	Freebase-150k	DBpedia-Music	YAGOc-195k
ALL	**0.904**	**0.843**	**0.911**
mwNN + TransE	0.893	0.842	0.909
mwNN + RESCAL	0.852	0.762	0.862
TransE + RESCAL	0.826	0.825	0.901
Best Single Predictor	(mwNN) 0.837	(TransE) 0.826	(TransE) 0.896

for the second setting, where the models solely exploited the Local Closed-World Assumption. *ALL* represents the ensemble consisting of TransE, RESCAL and mwNN and the beneath three models represent the ensembles that combine all possible pairs of models, e.g. mwNN + TransE represents the ensemble consisting of mwNN and TransE. *Best Single Predictor* represents the best model out of TransE, RESCAL or mwNN on the same link-prediction task. Which of the three models had the best AUPRC score is shown in the brackets next to the corresponding score in that row. d is the chosen dimension of the embedding vector or the rank of the factorization in case of RESCAL (e.g. $\mathbf{Model} \leftarrow \mathbf{d} = \mathbf{100}$) indicates that all models were trained with a fixed embedding dimension of 100).

4.3 Type-Constrained Ensembles Improve Prediction Quality Especially with Low-Dimensional Embeddings

From Table 2 it can be observed that the ensemble consisting of all three models (with type-constraints) is clearly outperforming the best single predictor on all datasets and with all different embedding dimensions (10,50,100). We observed the largest improvements on the Freebase-150k dataset, where the ensemble increases the AUPRC score from 0.775 to 0.846 with an embedding dimension of 10 and from 0.837 to 0.904 with an embedding dimension of 100. In the other two datasets large improvements for lower dimensional latent embedding vectors ($d = 10$, 11% on DBpedia-Music and 5% on YAGOc-195k) can be observed, but for higher dimensional embeddings ($d = 50$ and $d = 100$) the improvements are decreasing or vanishing (YAGOc-195k).

The improvements observed on the really low embedding dimension of 10 are of special interest. In a Web-Scale application of these algorithms it is of high interest to have meaningful embeddings in a very low dimensional latent space, because higher dimensional representation can lead to long or even intractable runtimes for model training and tuning of the different algorithms. It can be observed that the ensemble consisting of TransE, RESCAL and mwNN with a embedding dimension of 10 reaches comparable link prediction results than the best single predictor with an embedding vector dimension of 100. On Freebase-150k dataset the ensemble reaches with $d = 10$ an AUPRC score of 0.846, on DBpedia-Music 0.815 and YAGOc-195k 0.883, where the best single predictor reaches at $d = 100$ 0.837, 0.826 and 0.896, respectively.

When it comes to the contribution of each single predictor in the ensemble, we observe that in case of the Freebase dataset all models are contributing to the superior performance of the ensemble, but TransE and mwNN are responsible for the biggest increase in AUPRC. For example with $d = 10$, TransE+mwNN achieves an AUPRC of 0.820 where mwNN+RESCAL reaches 0.795 and TransE+RESCAL 0.757. On the DBpedia-Music and YAGOc-195k dataset, RESCAL does not add any significant value to the ensemble, e.g. for $d = 50$ mwNN+TransE has the highest AUPRC score of the pairwise ensembles reaching already 0.825 on DBpedia-Music and 0.900 on YAGOc-195k, where the maximum performance of the complete ensemble (ALL) lies at 0.827 and 0.902.

As a final remark, we could observe from the results shown in Table 2 that RESCAL or mwNN best complement with TransE. The combination of mwNN and RESCAL generally shows less improvements in AUPRC compared to the best single predictor performance of those two (compare with results in [8]), indicating that these two models learn more similar patterns.

4.4 Link-Prediction Quality also Improves by Combining the Single Predictors under a Local Closed-World Assumption

The results for the LCWA ensemble are shown in Table 3. The ensemble improves the AUPRC score compared to the best single predictor from 15% ($d = 10$) to 9% ($d = 100$) for the Freebase-150k dataset. Also, all predictors (RESCAL, TransE, mwNN) contribute to the performance of the ensemble, since the best pairwise ensemble achieves a significantly lower AUPRC score in this case. TransE+RESCAL achieves 0.763 ($d = 10$), 0.876 ($d = 50$) and 0.899 ($d = 100$), where the full ensemble achieves 0.775, 0.886, 0.909. On the DBpedia-Music dataset the ensemble only improves the best single predictor for very low dimensional embeddings ($d = 10$) from 0.719 to 0.787. The ensemble does not improve the best single predictor for a vector dimension of 50 and 100 in this dataset. The ensemble constantly improves the best single-predictor on the YAGOc-195k dataset of about 0.03 to 0.04 in AUPRC for all embedding vector dimensions. We also see small improvements of the full ensemble opposed to the best ensemble consisting of TransE and mwNN. As in case of the type-constrained ensembles,

Table 3. AUPRC results on datasets, exploiting the **Local Closed-World Assumption** in the models. **Model←*d*** indicates the dimensionality of the latent embeddings used by the models.

	Dataset		
Model←*d*=10	Freebase-150k	DBpedia-Music	YAGOc-195k
ALL	**0.775**	**0.787**	**0.825**
mwNN + TransE	0.729	0.780	0.820
mwNN + RESCAL	0.649	0.661	0.679
TransE + RESCAL	0.763	0.746	0.806
Best Single Predictor	(TransE) 0.671	(TransE) 0.719	(TransE) 0.790
Model←*d*=50	Freebase-150k	DBpedia-Music	YAGOc-195k
ALL	**0.886**	**0.841**	**0.899**
mwNN + TransE	0.854	**0.841**	0.890
mwNN + RESCAL	0.820	0.661	0.828
TransE + RESCAL	0.876	0.746	0.878
Best Single Predictor	(TransE) 0.806	(TransE) 0.839	(TransE) 0.861
Model←*d*=100	Freebase-150k	DBpedia-Music	YAGOc-195k
ALL	**0.909**	0.844	**0.900**
mwNN + TransE	0.884	0.844	0.890
mwNN + RESCAL	0.852	0.734	0.847
TransE + RESCAL	0.899	0.845	0.886
Best Single Predictor	(TransE) 0.831	(TransE) **0.848**	(TransE) 0.872

we can also observe from Table 2, that mwNN and RESCAL best complement with TransE.

5 Conclusion

In this work we showed that the predictions of three leading latent variable models for link-prediction in knowledge-graphs are indeed complementary to each other, which can be exploited in an ensemble solution. In our experiments we observed that especially TransE learns substantially different aspects of the data than RESCAL and mwNN. RESCAL and mwNN on the other hand are more similar to each other, even though these two models differ in various aspects. We further showed that an ensemble consisting of all three methods brings substantially higher prediction quality on all used datasets and all settings when the models need to exploit a very low dimensional embedding space ($d = 10$). The LCWA can also be exploited in the ensemble when the type-constraints for properties are absent or fuzzy. On the DBpedia-Music and YAGOc-195 dataset we observed that with a higher dimensional latent embedding space the improvements become less significant.

References

1. C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. Dbpedia - a crystallization point for the web of data. *Web Semant.*, 7(3):154–165, 2009.
2. K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: A collaboratively created graph database for structuring human knowledge. In *SIGMOD*, pages 1247–1250, 2008.
3. A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. In *NIPS 26*, pages 2787–2795. 2013.
4. K.-W. Chang, W.-t. Yih, B. Yang, and C. Meek. Typed tensor decomposition of knowledge bases for relation extraction. In *EMNLP*, pages 1568–1579, 2014.
5. A. Garcia-Duran, A. Bordes, and N. Usunier. Effective blending of two and three-way interactions for modeling multi-relational data. In *ECML–PKDD*, pages 434–449, 2014.
6. M. A. Hasan, V. Chaoji, S. Salem, and M. Zaki. Link prediction using supervised learning. In *SDM workshop on Link Analysis, Counterterrorism and Security*, 2006.
7. J. Hoffart, F. M. Suchanek, K. Berberich, E. Lewis-Kelham, G. de Melo, and G. Weikum. Yago2: Exploring and querying world knowledge in time, space, context, and many languages. In *WWW*, pages 229–232, 2011.
8. D. Krompaß, S. Baier, and V. Tresp. Type-constrained representation learning in knowledge graphs. In *ISWC*, 2015.
9. D. Krompaß, M. Nickel, and V. Tresp. Large-scale factorization of type-constrained multi-relational data. In *DSAA'2014*, 2014.
10. D. Krompaß, M. Nickel, and V. Tresp. Querying factorized probabilistic triple databases. In *ISWC*, 2014.
11. N. Lao, T. Mitchell, and W. W. Cohen. Random walk inference and learning in a large scale knowledge base. In *EMNLP*, pages 529–539, 2011.
12. Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, 2015.
13. M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich. A review of relational machine learning for knowledge graphs: From multi-relational link prediction to automated knowledge graph construction. In *arXiv:1503.00759*, 2015.
14. M. Nickel, V. Tresp, and H.-P. Kriegel. A three-way model for collective learning on multi-relational data. In *ICML*, pages 809–816, 2011.
15. M. Nickel, V. Tresp, and H.-P. Kriegel. Factorizing yago: scalable machine learning for linked data. In *WWW*, pages 271–280, 2012.
16. J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, pages 61–74, 1999.
17. M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62(1-2):107–136, 2006.
18. R. Socher, D. Chen, C. D. Manning, and A. Y. Ng. Reasoning With Neural Tensor Networks For Knowledge Base Completion. In *NIPS 26*. 2013.
19. Z. Wang, J. Zhang, J. Feng, and Z. Chen. Knowledge graph embedding by translating on hyperplanes. In *AAAI*, July 2014.
20. D. Xin L., et al. Knowledge vault: A web-scale approach to probabilisitic knowledge fusion. In *SIGKDD*, pages 601–610, 2014.
21. B. Yang, W.-t. Yih, X. He, J. Gao, and L. Deng. Embeddings entities and relations for learning and inference in knowledge bases. In *arXiv:1412.6575*.