Probabilistic Logic Graph Attention Networks for Reasoning

L Vivek Harsha Vardhan*
School of Computing
National University of Singapore
Singapore
harsha@comp.nus.edu.sg

Guo Jia*
School of Computing
National University of Singapore
Singapore
guojia@u.nus.edu

Stanley Kok
School of Computing
National University of Singapore
Singapore
skok@comp.nus.edu.sg

ABSTRACT

Knowledge base completion, which involves the prediction of missing relations between entities in a knowledge graph, has been an active area of research. Markov logic networks, which combine probabilistic graphical models and first order logic, have proven to be effective on knowledge graph tasks like link prediction and question answering. However, their intractable inference limits their scalability and wider applicability across various tasks. In recent times, graph attention neural networks, which capture features of neighbouring entities, have achieved superior results on highly complex graph problems like node classification and link prediction. Combining the best of both worlds, we propose Probabilistic Logic Graph Attention Network (pGAT) for reasoning. In the proposed model, the joint distribution of all possible triplets defined by a Markov logic network is optimized with a variational EM algorithm. This helps us to efficiently combine first-order logic and graph attention networks. With the goal of establishing strong baselines for future research on link prediction, we evaluate our model on various standard link prediction benchmarks, and obtain competitive results.

CCS CONCEPTS

• **Computing methodologies → Probabilistic reasoning**; *Artificial intelligence*; **Knowledge representation and reasoning**;

KEYWORDS

Graph attention networks, Markov logic networks, Link prediction, Knowledge graphs

ACM Reference Format:
L Vivek Harsha Vardhan, Guo Jia, and Stanley Kok. 2020. Probabilistic Logic Graph Attention Networks for Reasoning. In *Companion Proceedings of the Web Conference 2020 (WWW '20 Companion), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3366424.3391265

1 INTRODUCTION

Knowledge graphs are used in recommendation systems [6], semantic search [1, 2] and question answering [28]. Knowledge graph reasoning, which is the task of inferring hidden relations between entities based on observed relations, is an active area of research. Markov logic networks [17] combine first-order logic and probabilistic graphical models. However, the computational intractability of Markov logic networks limit their performance and applicability to various tasks. In another line of work, graph convolutional networks [9] (e.g.,R-GCN [19]) have shown their applicability to relational data. Specifically, attention-based knowledge graph embeddings [11] have shown significantly superior performance on tasks like link prediction. But they do not consider logic rules and therefore miss the knowledge that can be inferred from them. Hence, we leverage the best of both worlds by combining Markov logic networks and graph attention neural networks via a variational EM algorithm [12]. In the E step, we use graph-attention-neural-network embeddings for inferring the unobserved triplets. In the M step, the weights of the rules of a Markov logic network are updated based on the observed triplets and the inferred triplets obtained from the aforementioned embeddings. Our proposed Probabilistic Graph Attention Network (pGAT) yields competitive results over the present state-of-the-art baselines on the FB15K-237 [8] and WN18RR [23] datasets for link prediction. Our contributions lie in proposing pGAT, and establishing strong baselines for future research to compare against.

2 RELATED WORK

First order logic has been extensively used for reasoning in the past [21, 26]. Markov logic networks [18], which combines logic rules and probabilistic graphical models, are very effective at reasoning but their inference remains intractable for large datasets like those typically used for knowledge base completion. This is due to their complicated graph structures and their underlying combinatorially-huge computational graphs.

Knowledge graph embedding methods [5, 20] are recent advancements that use low dimensional embeddings of entities and relations to represent their semantics and reason about them. These methods use various scoring functions to model different rule patterns. In TransE [5], each relation is represented as a translational vector to model composition rules and inverse rules effectively. RotateE [20] models each relation as a rotation in a complex space. Convolution neural networks (CNN) based models, like ConvE [8] and ConvKB [13], use convolution filters on knowledge graphs for link prediction. Recent works like KBAT [11] show that graph attention networks lead to more effective knowledge graph embeddings. pLogicNet [16] uses Markov logic networks with neural networks for inference. However, pLogicNet's vanilla neural network does not effectively model relational data because it considers each triplet independently, and thus neglects the interaction among triplets. Some recent works like Graph Markov Neural Networks

*Both authors contributed equally to this research.

WWW '20 Companion, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7024-0/20/04.
https://doi.org/10.1145/3366424.3391265

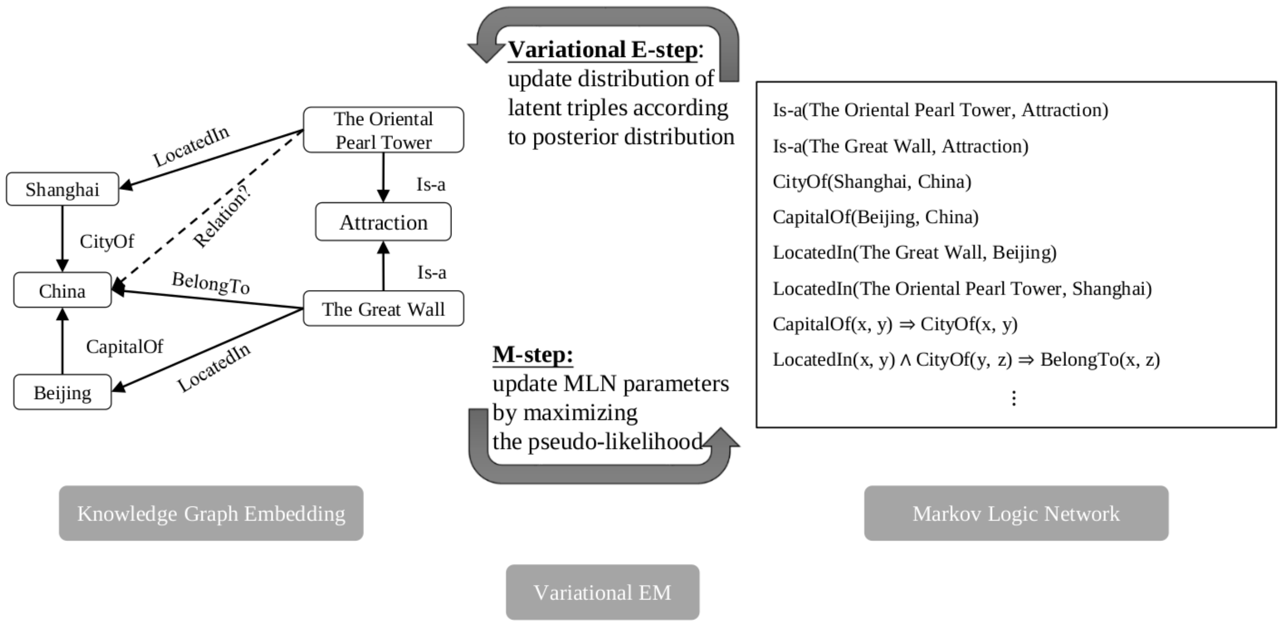

Figure 1: Illustration of our work, pGAT.

[15] use statistical relational learning and graph neural networks for the task of semi-supervised node classification.

3 APPROACH

We first introduce the knowledge graph embedding model employed in our framework. Then we elaborate on how the learned graph-attention-based embeddings and the parameter learning of Markov logic networks benefit each other in an iterative manner.

A knowledge graph $\mathcal{KG}$ is represented as a set of relational triples, $\mathcal{KG} = \{(e_h, r_k, e_t)\}$, where $e_h, e_t \in \mathcal{E}$, $r_k \in \mathcal{R}$, and $\mathcal{E}$ and $\mathcal{R}$ are a set of entities and a set of relations respectively. $\mathcal{T}_O$ and $\mathcal{T}_U$ respectively denote the set of observed triples and unobserved triples. Each triple $(e_h, r_k, e_t) \in \mathcal{T}_O$ indicates that there exists a directed relation r_k from head entity e_h to tail entity e_t ($e_h \neq e_t$).

3.1 Graph Attention-Based Embeddings

A recent model KBAT [11] surpassed the performances of several state-of-the-art knowledge-embedding models. We choose KBAT as our knowledge embedding component because its embeddings provide rich semantic information about the relations between entities. This allows for more accurate inference for the subsequent parameter learning of logical rules.

The inputs to the attention layer in KBAT are a node embedding matrix $\mathbf{X} \in \mathbb{R}^{N \times d_e}$, and a relation embedding matrix $\mathbf{Q} \in \mathbb{R}^{M \times d_r}$, where N and M are the total numbers of entities and relations respectively, and d_e and d_r are the dimensions of the embedding vectors for each entity and each relation respectively.

The feature vector of each triple (e_i, r_k, e_j) is represented as $\mathbf{t}_{ijk} = \mathbf{W}_t[\mathbf{x}_i || \mathbf{x}_j || \mathbf{q}_k]$, where $\mathbf{W}_t$ represents the weight matrix of a linear transformation over the concatenation of entity embeddings $\mathbf{x}_i, \mathbf{x}_j$, and relation embedding $\mathbf{q}_k$.

The vector $\mathbf{t}_{ijk}$ is used to produce a corresponding attention score $e_{ijk} = LeakyRelu(\mathbf{W}_e \mathbf{t}_{ijk})$. The attention score e_{ijk} is normalized through a $softmax$ function over all entities and relations in the neighborhood of entity e_i to give the normalized value α_{ijk}. The new embedding vector for entity i is computed as the weighted sum of all its connected triple representations weighted by the normalized attention values. A multi-head attention mechanism is used to compute the embedding vectors for entities. The final entity embedding vector is obtained by concatenating the entity embedding vectors from multiple attention mechanisms, and averaging over the vectors as follows:

$$\hat{\mathbf{x}}_i = \sigma\left(\frac{1}{C}\sum_{c=1}^{C}\sum_{j \in \mathcal{N}_i}\sum_{k \in \mathcal{Q}_{ij}} \alpha^c_{ijk}\mathbf{t}_{ijk}\right),$$

where $\mathcal{N}_i, \mathcal{Q}_{ij}$ represent all connected nodes and relations of entity i respectively, and C is the number of head attentions. The final embedding matrices for entities $\tilde{\mathbf{X}}$, and for relations $\tilde{\mathbf{Q}}$ are computed as:

$$\tilde{\mathbf{X}} = \mathbf{W}^x\mathbf{X} + \hat{\mathbf{X}}, \text{ and}$$
$$\tilde{\mathbf{Q}} = \mathbf{W}^q\mathbf{Q},$$

where $\mathbf{W}^x, \mathbf{W}^q$ are both parameters of linear transformations. This model is trained by minimizing the pairwise ranking loss:

$$\mathcal{L} = \sum_{s_{ijk} \in \mathcal{T}_O}\sum_{s'_{ijk} \in \mathcal{T}'_O} \max\{\theta + d_{s_{ijk}} - d_{s'_{ijk}}, 0\},$$

where θ is a margin hyper-parameter, and the distance function of each triple is given by $d_{s_{ijk}} = \|\tilde{\mathbf{x}}_i + \tilde{\mathbf{q}}_k - \tilde{\mathbf{x}}_j\|_1$ [5]. $\mathcal{T}'_O$ is the set

Table 1: Results of link prediction on test sets of FB15K-237 and WN18RR respectively. The best scores are in bold. The second best scores are underlined.

Method	MR	MRR	H@1	H@3	H@10	MR	MRR	H@1	H@3	H@10
TransE [5]	323	0.279	19.8	37.6	44.1	2300	0.243	4.27	44.1	53.2
DistMult [27]	512	0.281	19.9	30.1	44.6	7000	0.444	41.2	47	50.4
ComplEx [24]	546	0.278	19.4	29.7	45	7882	0.449	40.9	46.9	50.4
RotatE [20]	185	0.297	20.5	32.8	48.0	3277	**0.470**	**42.2**	48.8	56.5
ConvE [8]	245	0.312	22.5	34.1	49.7	4464	0.456	41.9	47	53.1
ConvKB [13]	216	0.289	19.8	32.4	47.1	**1295**	0.265	5.82	44.5	55.8
R-GCN[19]	600	0.164	10	18.1	30	6700	0.123	20.7	13.7	8
KBAT [11]	204	0.431	35.5	46.2	57.8	1970	0.431	35.2	47.3	57.4
BLP [7]	1985	0.092	6.2	9.8	15.0	12051	0.254	18.7	31.3	35.8
MLN [17]	1980	0.098	6.7	10.3	16.0	11549	0.259	19.1	32.2	36.1
pLogicNet [16]	173	0.330	23.1	36.9	52.8	3436	0.230	1.5	41.1	53.1
pLogicNet* [16]	**173**	0.332	23.7	36.7	52.4	3408	0.441	39.8	44.6	53.7
pGAT	181	**0.457**	**37.7**	**49.4**	**60.9**	1868	0.459	39.5	**48.9**	**57.8**

of negative triples created by randomly replacing the head or tail entity of triples in $\mathcal{T}_O$. We employ ConvKB [13] as our decoder after updating entity embeddings and relation embeddings, and train it using soft-margin loss [25]. More details about the decoder are found in [13, 25].

3.2 Parameter learning of MLN

A Markov logic network (MLN) [17] consists of a set of weighted first-order logic formulas, which can be viewed as templates for constructing Markov networks. Given a knowledge graph $\mathcal{KG}$, the joint probability of all relational triples is given by

$$p(\mathbf{s}) = \frac{1}{Z}\exp\left(\sum_{l=1}^{L} w_l n_l(\mathbf{s})\right),$$

where $\mathbf{s}$ represents triple $(e_i, r_k, e_j) \in \mathcal{KG}$, L is the total number of logical formulas in the MLN, $n_l(\mathbf{s})$ is the number of true groundings of the l^{th} logical rule (that has $\mathbf{s}$ as is consequent) according to the truth value of $\mathcal{KG}$'s triples, and w_l is the weight parameter of the l^{th} logical rule. We follow the training regime which uses a variational EM algorithm [12]. Due to the intractability of directly optimizing the joint probability distribution, we instead optimize the following evidence lower bound (ELBO) of its log-likelihood function.

$$\log p(\mathbf{s}_o) \geq \log p(\mathbf{s}_o) - KL[q(\mathbf{s}_u) \| p(\mathbf{s}_u|\mathbf{s}_o)]$$
$$= \int q(\mathbf{s}_u) \log p(\mathbf{s}_o, \mathbf{s}_u) \mathrm{d}\mathbf{s}_u - \int q(\mathbf{s}_u) \log q(\mathbf{s}_u) \mathrm{d}\mathbf{s}_u,$$

where KL denotes the KL divergence [10], and $q(\cdot)$ represents the variational distribution of unseen triples $\mathbf{s}_u$. Equality in the above expression holds when $q(\mathbf{s}_u) = p(\mathbf{s}_u|\mathbf{s}_o)$. In the variational E-step, we fix $p(\cdot)$ and optimize $q(\cdot)$ by inferring the true posterior distribution with mean-field approximation [14], which is based on the learned embeddings of the knowledge embedding model (KBAT) we have trained, i.e.,

$$q(\mathbf{s}_u) = \prod_{\mathbf{s}_u \in \mathcal{T}_U} Pr(s_u | f_{score}(s_u)),$$

$Pr(\cdot)$ measures the true probability of triples with the score funtion f_{score}, which is computed on entity and relation embeddings. The score function measures the plausibility of triples. Through minimizing the KL divergence between $q(\mathbf{s}_u)$ and the true posterior distribution $p(\mathbf{s}_u|\mathbf{s}_o)$, the optimal $q(\mathbf{s}_u)$ is computed as

$$\log q(\mathbf{s}_u) = \mathbb{E}_{q(\mathbf{s}_{u,MB})}[\log p(\mathbf{s}_u|\mathbf{s}_{u,MB})] + \text{const},$$

where $\mathbf{s}_{u,MB}$ is the Markov blanket of $\mathbf{s}_u$. If there exists any unseen triple in $\mathbf{s}_{u,MB}$, we replace it with a sample from the potential distribution inferred by the knowledge graph embedding model. To further optimize the objective, we enhance the knowledge embedding model by updating its training dataset with added unseen triples, which are predicted by the MLN. In the M-step, we fix $q(\cdot)$ and update $p(\cdot)$ by maximizing the pseudo-likelihood [3] as follows:

$$\mathcal{L}_{pseudo} = \mathbb{E}_{q(\mathbf{s}_u)}\Big[\sum_{\mathbf{s}_u \in \mathcal{T}_U, \mathbf{s}_o \in \mathcal{T}_O} \log p(\mathbf{s}_u, \mathbf{s}_o)\Big]$$
$$= \mathbb{E}_{q(\mathbf{s}_u)}\Big[\sum_{\mathbf{s} \in \{\mathcal{T}_O \cup \mathcal{T}_U\}} \log p(\mathbf{s}|\mathbf{s}_{MB})\Big],$$

Similar to the variational E-step, we fill those unseen triples in the Markov blanket with samples obtained from our knowledge embedding model. Therefore, by alternating between the variational E-step and an M-step, this framework allows knowledge sharing

Table 2: Hyperparameters for Graph Attention Embedding model.

Dataset	Weight decay	Epochs	Negative-ratio	Learning-rate	Dropout	Leaky Relu	Heads	Final-dimensions
FB15K-237	$1e^{-5}$	3000	2	$1e^{-3}$	0.3	0.2	2	200
WN18RR	$5e^{-5}$	3600	2	$1e^{-3}$	0.3	0.2	2	200

Table 3: Hyperparameters for Decoder (ConvKB).

Dataset	Weight decay	Epochs	Negative-ratio	Learning-rate	Dropout	Filters
FB15K-237	$1e^{-5}$	200	40	$1e^{-3}$	0.3	50
WN18RR	$1e^{-5}$	200	40	$1e^{-3}$	0	500

between the knowledge graph embedding model and the MLN as in shown in Figure 1.

4 EXPERIMENTS

4.1 Datasets

We evaluated our model on the task of link prediction knowledge graphs. We report the results of our work on two benchmark datasets FB15K-237 from Freebase [4] and WN18RR from Wordnet [5], which are created to resolve the reversible relation problem in their respective source datasets.

4.2 Evaluation Metrics

We remove a head entity or tail entity and predict the resulting triple using our proposed model. The average scores from replacing head and replacing tail are reported. Other baselines results are taken from their official code implementations and KBAT [11]. We evaluated our models in a filtered setting following previous works [11]. We report mean reciprocal rank (MRR), mean rank (MR) and the proportion of correct entities in the top N ranks (Hits@N) for N = 1, 3, and 10.

Table 4: Hyperparameters for Markov Logic Network.

Dataset	Rule-threshold	Triplet-threshold
FB15K-237	0.6	0.7
WN18RR	0.1	0.4

4.3 Implementation Details

We used TransE [5] to initialise the knowledge graph embedding before training. The knowledge graph embedding is used for inference at test time. The candidate rules for Markov logic network are generated using a brute force method where we search for all possible composition rules, inverse rules, symmetric rules and subrelation rules in the observed triplets. We use an accuracy threshold above which we select a rule as a candidate rule in our Markov logic network. We also use a probability threshold above which we add a triplet to the training set for knowledge graph embeddings. We use the Adam optimizer for training knowledge graph embeddings. The hyper-parameters of our model are in Table 2, Table 3 and Table 4.

4.4 Baselines

We compare our proposed pGAT model against knowledge graph embedding methods (TransE, DistMult, ComplEx, RotateE [5, 20, 24, 27]) , convolution-based knowledge graph embedding methods (ConvE [8] and ConvKB [13]), and graph-neural-networks-based knowledge graph embedding methods (R-GCN [19] and KBAT [11]). We also compare against Markov logic networks (MLN) [17], Bayesian logic programs (BLP) [7], pLogicNet, and pLogicNet*[16].

5 RESULTS AND DISCUSSION

The results on two datasets are presented in Table 1. Our proposed pGAT model outperforms all the baseline methods on the FB15K-237 dataset, achieving 0.457 for MRR and 37.7 for HITS@1 . We are one of the top two performers on the WN18RR dataset for most of the metrics. The good empirical results of our PGAT model bears out the efficacy of incorporating domain knowledge in the form of Markov logic rules and using graph attention networks to leverage neighbourhood information at various distances.

6 CONCLUSION

We propose pGAT, which combines Markov logic networks and graph attention networks, for the task of link prediction in knowledge graphs. Our work establishes strong baselines for future work on knowledge base completion. As future work, we plan to extend our model to inductive settings [22], and explore efficient ways to find rules for the Markov logic network. This research is partly funded by MOE AcRF Tier 1 grant (R -253-000-146-133) to Stanley Kok.

REFERENCES

[1] Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2010. Global Learning of Focused Entailment Graphs. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Uppsala, Sweden, 1220–1229. https://www.aclweb.org/anthology/P10-1124

[2] Jonathan Berant and Percy Liang. 2014. Semantic Parsing via Paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Baltimore, Maryland, 1415–1425. https://doi.org/10.3115/v1/P14-1133

[3] Julian Besag. 1975. Statistical Analysis of Non-Lattice Data. *Journal of the Royal Statistical Society. Series D (The Statistician)* 24, 3 (1975), pp. 179–195.

[4] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge *(SIGMOD '08)*. ACM, New York, NY, USA, 1247–1250. https://doi.org/10.1145/1376616.1376746

[5] 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 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 2787–2795. http://papers.nips.cc/paper/5071-translating-embeddings-for-modeling-multi-relational-data.pdf

[6] Robin Burke. 2000. Knowledge-based recommender systems. In *Encyclopedia of Library and Information Systems*. Vol. 69. Marcel Dekker, New York, NY, USA, 180–200. http://www.cs.pitt.edu/~mrotaru/comp/rs/Burke%20ELIS%202000.pdf

[7] Kersting K. De Raedt L. 2008. Probabilistic Inductive Logic Programming. In *Probabilistic Inductive Logic Programming.*, Kersting K. Muggleton S. De Raedt L., Frasconi P. (Ed.). Springer, Berlin, Chapter 1, 1–27.

[8] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2D Knowledge Graph Embeddings. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17366

[9] Thomas N. Kipf and Max Welling. 2016. Semi-Supervised Classification with Graph Convolutional Networks. *CoRR* abs/1609.02907 (2016). arXiv:1609.02907 http://arxiv.org/abs/1609.02907

[10] Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics* 22, 1 (1951), 79–86.

[11] 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*. Association for Computational Linguistics, Florence, Italy, 4710–4723. https://doi.org/10.18653/v1/P19-1466

[12] Radford M. Neal and Geoffrey E. Hinton. 1998. *A View of the Em Algorithm that Justifies Incremental, Sparse, and other Variants*. Springer Netherlands, Dordrecht, 355–368. https://doi.org/10.1007/978-94-011-5014-9_12

[13] Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung. 2018. A Novel Embedding Model for Knowledge Base Completion Based on Convolutional Neural Network. In *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. 327–333.

[14] Manfred Opper and David Saad. 2001. *Advanced mean field methods: Theory and practice*. MIT press.

[15] Meng Qu, Yoshua Bengio, and Jian Tang. 2019. GMNN: Graph Markov Neural Networks. arXiv:cs.LG/1905.06214

[16] Meng Qu and Jian Tang. 2019. Probabilistic Logic Neural Networks for Reasoning. arXiv:cs.LG/1906.08495

[17] Matthew Richardson and Pedro Domingos. 2006. Markov Logic Networks. *Mach. Learn.* 62, 1-2 (Feb. 2006), 107–136. https://doi.org/10.1007/s10994-006-5833-1

[18] Matthew Richardson and Pedro Domingos. 2006. Markov logic networks. *Machine Learning* 62, 1 (01 Feb 2006), 107–136. https://doi.org/10.1007/s10994-006-5833-1

[19] Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2017. Modeling Relational Data with Graph Convolutional Networks. arXiv:stat.ML/1703.06103

[20] 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*. https://openreview.net/forum?id=HkgEQnRqYQ

[21] Ben Taskar, Pieter Abbeel, and Daphne Koller. 2012. Discriminative Probabilistic Models for Relational Data. arXiv:cs.LG/1301.0604

[22] Komal K. Teru and William L. Hamilton. 2019. Inductive Relation Prediction on Knowledge Graphs. arXiv:cs.LG/1911.06962

[23] Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*. Association for Computational Linguistics, Beijing, China, 57–66. https://doi.org/10.18653/v1/W15-4007

[24] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International Conference on Machine Learning (ICML)*, Vol. 48. 2071–2080.

[25] Xiaobo Wang, Shifeng Zhang, Zhen Lei, Si Liu, Xiaojie Guo, and Stan Z. Li. 2018. Ensemble Soft-Margin Softmax Loss for Image Classification. arXiv:cs.CV/1805.03922

[26] Michael P. Wellman, John S. Breese, and Robert P. Goldman. 1992. From knowledge bases to decision models. *The Knowledge Engineering Review* 7, 1 (1992), 35–53. https://doi.org/10.1017/S0269888900006147

[27] Bishan Yang, Scott Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *Proceedings of the International Conference on Learning Representations (ICLR) 2015* (proceedings of the international conference on learning representations (iclr) 2015 ed.). https://www.microsoft.com/en-us/research/publication/embedding-entities-and-relations-for-learning-and-inference-in-knowledge-bases/

[28] Xuchen Yao and Benjamin Van Durme. 2014. Information Extraction over Structured Data: Question Answering with Freebase. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Baltimore, Maryland, 956–966. https://doi.org/10.3115/v1/P14-1090