Neural-Symbolic Commonsense Reasoner with Relation Predictors

Farhad Moghimifar[1] and **Lizhen Qu**[2] and **Yue Zhuo**[3]
Gholamreza Haffari[2] and **Mahsa Baktashmotlagh**[1]
[1]The School of ITEE, The University of Queensland, Australia
[2]Faculty of Information Technology, Monash University, Australia
[3]School of CSE, The University of New South Wales, Australia
{f.moghimifar,m.baktashmotlagh}@uq.edu.au
firstname.lastname@monash.edu, terry.zhuo@unsw.edu.au

Abstract

Commonsense reasoning aims to incorporate sets of commonsense facts, retrieved from Commonsense Knowledge Graphs (CKG), to draw conclusion about ordinary situations. The dynamic nature of commonsense knowledge postulates models capable of performing multi-hop reasoning over new situations. This feature also results in having large-scale sparse Knowledge Graphs, where such reasoning process is needed to predict relations between new events. However, existing approaches in this area are limited by considering CKGs as a limited set of facts, thus rendering them unfit for reasoning over new unseen situations and events. In this paper, we present a neural-symbolic reasoner, which is capable of reasoning over large-scale dynamic CKGs. The logic rules for reasoning over CKGs are learned during training by our model. In addition to providing interpretable explanation, the learned logic rules help to generalise prediction to newly introduced events. Experimental results on the task of link prediction on CKGs prove the effectiveness of our model by outperforming the state-of-the-art models.

1 Introduction

Commonsense reasoning refers to the ability of capitalising on commonly used knowledge by most people, and making decisions accordingly (Sap et al., 2020). This process usually involves combining multiple commonsense facts and beliefs to draw a conclusion or judgement (Lin et al., 2019). While human trivially performs such reasoning, current Artificial Intelligence models fail, mostly due to challenges of acquiring relevant knowledge and forming logical connections between them.

Recent attempts in empowering machines with the capability of commonsense reasoning are mostly centred around large-scale Commonsense Knowledge Graphs (CKG), such as ATOMIC and ConceptNet (Sap et al., 2019; Speer et al., 2017). Unlike conventional Knowledge Graphs (KG), CKGs usually contain facts about arbitrary phrases. For instance, “PersonX thanks PersonY" is connected to “ To express gratitude" via the link “ because X wanted". This non-canonicalised free-form text representation has resulted in having conceptually related nodes with different representation, which forms *large sparse* CKGs (Malaviya et al., 2020). Therefore, established reasoning models on conventional KGs perform poorly on CGKs (Yang et al., 2014; Sun et al., 2018; Dettmers et al., 2018; Minervini et al., 2020). In addition, the nature of commonsense reasoning encourages dynamic CKGs, where new sets of facts and phrases are introduced frequently. Most existing models in this realm are based on a static set of facts and phrases, which results in poor generalisation (Malaviya et al., 2020; Shang et al., 2019). Nevertheless, the inference process in existing approaches is like a *black box*, where internal behaviour of the model is hardly interpretable.

To overcome these limitations, we propose a neural-symbolic reasoning model based on backward-chaining. While traditional theorem proving algorithms (Bratko, 2001) work based on a set of predefined rules and unification over discrete symbols, we leverage a continuous relaxation of weak unification and a rule learner module. The weak unification over continuous embedding representation helps to address the challenges of unseen sparsity of CKGs. The rule learner module, in addition to providing interpretability, is used to generalise prediction to unseen data points to mitigate the problem of *large-scale dynamic* CKGs. The experiments on the task of link prediction confirm the superiority of our model, by a margin of up to 22 points, over the state-of-the-art models.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 797–802
August 1–6, 2021.

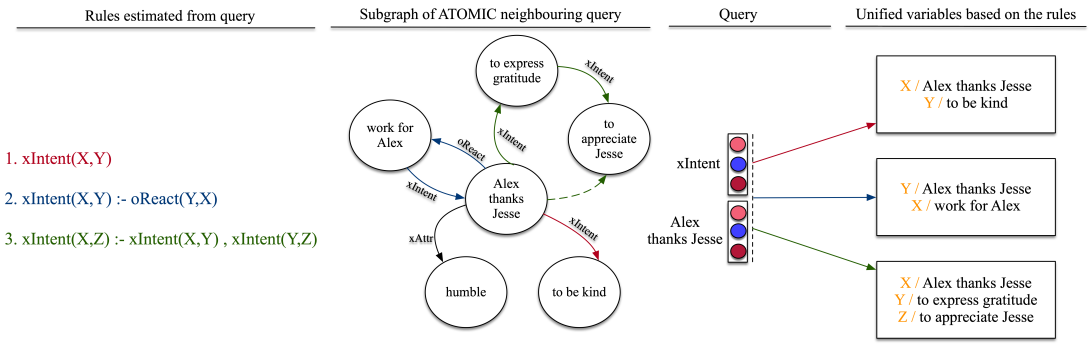

Figure 1: A visual representation of rules and new relations estimated by our model for a sample query, *xIntent(Alex thanks Jesse, ?)*. Based on the subject of the query, a subgraph of ATOMIC is retrieved for the reasoning process (middle). Sets of rules estimated from relation of the query is generated using our proposed rule creation module (left). Based on our reasoning model, the answers to query are predicted by unification module (right).

2 Related Works

Recent approaches in knowledge base completion task have mostly relied on a graph and entity-relation embedding methods (Yang et al., 2014; Dettmers et al., 2018). In these approaches, entities and relations are embedded in a complex space, and using a scoring function plausibility of a triple is estimated (Bordes et al., 2013; Trouillon et al., 2016; Sun et al., 2018). In addition to node embedding, graph embedding methods have also been used to capture the structural complexity of knowledge bases (Schlichtkrull et al., 2018; Shang et al., 2019). Language generative models also have been applied on knowledge bases in order to use the rich information of pre-trained models to address CKG completion task (Bosselut et al., 2019; Moghimifar et al., 2020). Malaviya et al. (2020) proposed a method based on using language models and graph networks to solve the problem of the sparsity of CKGs, by taking structural and contextual characteristics of CKGs into account. However, the aforementioned models are highly dependant on training on a set of static entities, and fail to perform when new triples are presented.

3 Our Approach

A CKG $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{E}$ is the set of edges in $\mathcal{G}$, consists of triples in form of $r(h, t)$, where $h, t \in \mathcal{N}$ are referred to as the head and the tail of the triple, and $r \in \mathcal{E}$ denotes their relation. The goal of the CKG completion task is to estimate probable t given a query $q = r(h, ?)$. As the target node may not pose a direct link to h via r, this task requires a model capable of multi-step reasoning.

Given a query $r_q(h_q, ?)$, we try to identify an implication rule and apply it to prove $r_q(h_q, t)$ for a target entity or event t. A rule $\mathcal{R}$ takes the form of $r_q(X, Z) :- r_0(X, Y_0), ..., r_k(Y_{k-1}, Z)$, where capitalised letters denote variables, $r_q(X, Z)$ is the rule head, and the rule body is a conjunction of atoms. We apply such a rule by unifying atoms with triples in the given CKG to obtain $r_q(h_q, t_k) :- r_0(h_0, t_0), r_1(t_0, t_1), ..., r_k(t_{k-1}, t_k)$, which entails $r_q(h_q, t_k)$. Since semantically equivalent/similar events or entities in a CKG often have different surface forms, we consider weak unification of an atom with a triple instead of only considering exact match of two atoms, a weak unification operator (Sessa, 2002) unifies two different symbols by measuring the similarity of their representations.

Given a query, we do not know the target rule in advance. As shown in the example in Fig. 1, we successively create a new rule by appending the body of the previous rule with an atom in the form of $r(t_{k-1}, X)$. Whenever such a new atom is added, we query the CKG to find triples as candidates of unification. This step enables reasoning on large scale KBs. In contrast, the prior works (Minervini et al., 2020; Ren and Leskovec, 2020) require comparison with each node in a CKG. After applying the weak unification operator to each of the triples, we find top$k$ most similar nodes and use each of the entity/event in the place of X to create a new atom for a new rule. The process is repeated until the maximal rule length is reached.

The above mentioned reasoning process is delivered by a a neural-symbolic reasoner. It consists of a query module, a weak unification operator, and a rule creation module.

Dataset	#Nodes	#Edges	Avg. In-degree	Density	Unseen Nodes	Unseen Edges	#Relations
ATOMIC	382823	785952	2.25	1.6e-5	38.36%	27.91%	9
ConceptNet-100k	80994	102400	1.25	9.0e-6	11%	8%	34

Table 1: Statistics on ATOMIC and ConceptNet-100k. Unseen Nodes is the ratio of the nodes in test set that are not in train set to all of the nodes in test set. Unseen edges is the ratio of edges where either the head or tail nodes are not in train set to the number of all edges in test set.

Query Given a rule with a rightmost atom $r_{k-1}(t_{k-1}, X)$ in the rule body, we send the representation of t_{k-1} as query to the given CKG to retrieve unification candidates. A node in a CKG is a word sequence. To support comparison of nodes w.r.t. their semantic similarities, we encode queries and nodes in a CKG with a pre-trained BERT (Devlin et al., 2019) into embeddings. To this end, a node is converted into $[CLS]+node+[SEP]$, and fed into the model, and we use the representation of $[CLS]$ token from the last layer of BERT as representation of node $node$. We apply FAISS[1] (Johnson et al., 2019) to index embeddings of an CKG, because it supports fast retrieval of k nearest neighbours of a dense vector. For each node v in the top$k$ list, we collect a set of triples $\mathcal{C}(v)$, which are all triples having v as the head in the CKG. As a result, we have k such sets and form a candidate set $\mathcal{C}$ by taking the union of them.

Weak Unification From a candidate set $\mathcal{C}$ we identify top$k$ most relevant triples to unify $r_{k-1}(t_{k-1}, X)$. First, we formulate a set of hypotheses $\mathcal{H}$ by replacing X with possible tails. In practice, we use all tails of the triples in C. Furthermore, we construct a bipartite graph between $\mathcal{C}$ and $\mathcal{H}$, in which an edge denotes the unification between a triple from $\mathcal{C}$ and another from $\mathcal{H}$. We measure unification scores by using cosine similarity and obtain an similarity matrix $\mathbf{U} \in \mathbb{R}^{|\mathcal{C}|\times|\mathcal{H}|}$. The final unification score of candidate triple i is computed by $\max_j \mathbf{U}_{ij}$. We keep only top$k$ highest scored candidate triples.

Rule creation Given the top$k$ highest scored candidate triples and a rule $\mathcal{R}_k$ with a rightmost atom $r_{k-1}(t_{k-1}, X)$, we create a new rule based on $\mathcal{R}_k$ for each triple k by substituting it for $r(t_{k-1}, X)$ and append another atom $r_k(t_k, X)$. The relation r_k is estimated by a relation predictor $\boldsymbol{f_\theta}(r_{k-1}, k)$, where both r_{k-1} and the current step k are mapped to the corresponding embeddings.

$$P_{\boldsymbol{\theta_f}}(r_k|r_{k-1}, k) = \sigma(\boldsymbol{f_\theta}([r_{k-1}; k]).\mathbf{W} + b) \quad (1)$$

[1]https://github.com/facebookresearch/faiss

where $\boldsymbol{\theta_f} := \{\mathbf{W}, b\}$ contains the Rule creation module's parameters, and σ is the sigmoid function. The relation predictor aims to generalise relation co-occurrence patterns in rules. We implement it by using a neural networks with two blocks of hidden layers, followed by a softmax layer. Each block is composed of a linear layer and a ReLU layer.

Given a query $r_q(h_q, ?)$, we initialise the first rule as $r_q(h_q, X)$. After reaching the pre-defined maximal rule length, we consider the score of a rule after unification as the lowest unification score associated with the rule, following (Sessa, 2002). We rank all rules by their scores and select the tails in the rule heads of the top$k$ highest scored rules as the results.

Another benefit of our reasoner is that humans can easily collect evidences to interpret reasoning results. The model can yield the rules and unified triples in a human-friendly format, which are generated at each step. In contrast, prior work (Malaviya et al., 2020) on commonsense reasoners produces only hard-to-understand distributed representations in intermediate steps.

Training We convert all the triples in $\mathcal{G}$ into a set of queries ($\mathcal{Q} = \{r_1(h_1, ?), r_2(h_2, ?), \ldots, r_n(h_n, ?), \}$), where each query of $r_i(h_i, ?)(i < n)$ is associated with a set of gold answers $\mathcal{T}_i = \{q_{i_1}, q_{i_2}, \ldots, q_{i_m}\}$. The goal of training our model is to learn the embedding representations by minimising a cross-entropy loss function ($\mathcal{L}_\theta$) on final scores associated with each estimated predictions and the set of gold answer:

$$\mathcal{L}_{\boldsymbol{\theta}} = -\sum_{q_{p_k}\in\mathcal{T}} \log(Pr(q_{p_k}|\mathcal{G}; \boldsymbol{\theta})) - \sum_{q_{p_k}\notin\mathcal{T}} \log(1 - Pr(Pr(q_{p_k}|\mathcal{G}; \boldsymbol{\theta}))) \quad (2)$$

where $\boldsymbol{\theta}$ denote all the parameters of our model. The relation predication module of our model is also trained by minimising loss in equation 2, where the relation embeddings are decoded by

	ConceptNet-100k				ATOMIC			
Model	MRR	HITS@1	HITS@3	HITS@10	MRR	HITS@1	HITS@3	HITS@10
DistMult	8.68	5.38	9.33	15.23	11.49	9.16	11.83	16.3
ComplEx	10.33	6.51	11.24	17.31	12.96	10.65	13.9	17.08
ConvE	16.55	10.19	18.79	28.08	9.04	7.05	9.42	12.74
RotatE	19.89	14.45	25.32	37.56	10.61	8.56	10.76	14.98
Malaviya et al.	43.60	39.33	49.41	66.58	23.43	20.54	24.1	27.43
Ours	**65.72**	**57.49**	**61.7**	**71.46**	**46.41**	**43.31**	**45.94**	**47.24**

Table 2: Results on CKG completion task, on ConceptNet-100K and ATOMIC.

alignment of the associated embedding and nearest predicate representation.

4 Experiments

To evaluate the performance of our model [2] in the task of CKG completion, in this section, we report the results of our model in comparison with the baselines.

Evaluation Metrics: Following previous works on Knowledge Base completion (Dettmers et al., 2018; Malaviya et al., 2020), we report the results of HITS and Mean Reciprocal Rank. Similar to Dettmers et al. (2018), when computing the scores for a gold target entity, we filter out all remaining valid entities. Furthermore, for each triple (h, r, t), the score is the average of scores measured from $(h, r, ?)$ and $(t, r^{-1}, ?)$.

Baselines For comparison, we report the performance of state-of-the-art models in CKG and KB completion. We compare our model to DistMult (Yang et al., 2014), ComplEx (Trouillon et al., 2016), ConvE (Dettmers et al., 2018), RotatE (Sun et al., 2018), and Malaviya (Malaviya et al., 2020). The first four models are high performance models in conventional KB completion, whereas the latter is proposed for CKG completion.

4.1 Datasets

ATOMIC [3] is a CKG consisting of commonsense facts in form of triples, based on *if-then* relations (Sap et al., 2019). This dataset consists of more 877K facts, and more than 300K entities.

ConceptNet-100K [4] is a subset of ConceptNet 5 (Speer et al., 2017), containing Open Mind Common Sense (OMCS) entries, introduced by (Li et al., 2016). This dataset contains general commonsense facts in form of triples.

In order to evaluate the performance of the models in dynamic CKG completion, we choose a subset of the test set of ATOMIC and ConceptNet-100K, where for any (h, r, t) either h or t is not seen by the model in the train set. Statistics on ATOMIC and ConceptNet-100k are provided in table 1. To train our model, each triple in form $r(h, t)$ in train set was also converted to $r^{-1}(t, h)$, to account for reverse relations as well. We have used the embedding size of 1024 for both node and relation embedding layer. To embed the nodes in CKGs, we have fine-tuned uncased BERT-Large (Devlin et al., 2019) for the objective of masked language model. For this purpose, a node is converted into $[CLS] + n_i + [SEP]$ and fed into BERT. The representation of the token $[CLS]$ from the last layer of BERT is then used as node n_i embedded representation. We used the maximum sequence of 128, and batch size of 64. Our relation predication module consists of two Linear layer. For all non-linearities in our model we have used ReLU. For optimisation purpose, SGD has been used, with staring learning rate of $10e-4$, and decay rate of 0.9, if the loss of development set does not decrease after each epoch. We set the maximum depth of three for reasoning process. We have trained the model for 200 epochs. Followed by Malaviya et al. (2020), we have trained all the baselines for 200 epochs. During training the models were evaluated on development set, every 10 and 30 epochs, for ConceptNet-100K and ATOMIC, respectively. The checkpoint with the highest MRR was then selected for testing.

4.2 Results

Table 2 summarises the results of the conducted experiment on ConceptNet-100K and ATOMIC. On ConceptNet-100K our proposed model outperforms the baselines by up to 22 points on MRR. The gap between our model and the second best model decrease as we move from HITS@1 to HITS@10.

[2] Code available at https://github.com/farhadmfar/commonsense_reasoner

[3] https://homes.cs.washington.edu/ msap/atomic/

[4] https://ttic.uchicago.edu/ kgimpel/commonsense.html

ATOMIC
`xIntent(X,Y):-xIntent(X,Z),xIntent(Z,Y)`
`xNeed(X,Y):-xReact(Y,X)`
`xIntent(X,Y):-oWant(Y,X)`
ConceptNet-100K
`causes(X,Y):-causes(X,Z),causes(Z,Y)`
`isa(X,Y):-partof(X,Z),isa(Z,Y)`
`relatedto(X,Y):-relatedto(X,Z),relatedto(Z,Y)`

Table 3: Examples of rules learned by our proposed relation prediction module.

This suggested that on contrary to the baselines our model performs better in estimating the probability of query with higher accuracy. On ATOMIC our model achieves a MRR of 46.41, which is 23 points higher than the second best model. As it can be seen from table 2, comparison of performance of different models on ConceptNet-100K and ATOMIC shows a noticeable drop in performance for models which rely on structural information of CKGs. This observation suggests that larger and sparser (lowest density) CKG are more challenging to reason over.

Table 3 provides examples of generated rules by our model on ATOMIC and ConceptNet-100k. On ATOMIC, the first rule is based on transition, and the second and third rules are inverse rules. Similarly, on ConceptNet-100K the first and third rules are transitive, and the second rule is a compositional rule. All provided rules are diverse and meaningful, and can be used for explaining the inference process of our model. For instance, consider a query of *xIntent(Alex drives Jesse there, ?)*. Based on first rule from Table 3, `X` is unified by *Alex drives Jesse there*, and `Z` is unified by *Alex helps Jesse* (from triples of ATOMIC). Then, the query is updated to *xIntent(Alex helps Jesse, ?)* and `Y` is unified by *to be of assistance* (from triples of ATOMIC), hence the answer to query. The path generated by this example is *Alex drives Jesse there* $\xrightarrow{\text{xIntent}}$ *Alex helps Jesse* $\xrightarrow{\text{xIntent}}$ *to be of assistance*. Therefore, two nodes are connected via a new link: *Alex drives Jesse there* $\xrightarrow{\text{xIntent}}$ *to be of assistance*.

Consider the following query from ConceptNet-100K, $HasProperty(novel, ?)$. Based on the relation of the query, our rule creator module can estimate the following rule: According to this rule,

```
HasProperty(X,Y):-IsA(X,Z),HasProperty(Z,Y)
```

`X` is unified by *novel*, and `Z` is unified by *book* (from triples of ConceptNet-100K). Then, the query is updated to *HasProperty(book, ?)* and `Y` is unified by *expensive* (from triples of ConceptNet-100K), resulting the answer to the query, by generating the following path: *novel* $\xrightarrow{\text{IsA}}$ *book* $\xrightarrow{\text{HasProperty}}$ *expensive*, hence *novel* $\xrightarrow{\text{HasProperty}}$ *expensive*.

5 Conclusion

In this work, we propose a neural-symbolic reasoning model over Commonsense Knowledge Graphs (CKGs). Our proposed model leverages a relation prediction module, which provides capability of multi-step reasoning. This ability, alongside weak unification, helps generalising our model to large-scale unseen data. We showed that our model yields state-of-the-art results when applied to large-scale sparse CKGs, and the inference step is interpretable.

References

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Neural Information Processing Systems (NIPS)*.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. Comet: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Ivan Bratko. 2001. *Prolog programming for artificial intelligence*. Pearson education.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*.

Xiang Li, Aynaz Taheri, Lifu Tu, and Kevin Gimpel. 2016. Commonsense knowledge base completion. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. Kagnet: Knowledge-aware graph

networks for commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2822–2832.

Chaitanya Malaviya, Chandra Bhagavatula, Antoine Bosselut, and Yejin Choi. 2020. Commonsense knowledge base completion with structural and semantic context. In *AAAI*.

Pasquale Minervini, Sebastian Riedel, Pontus Stenetorp, Edward Grefenstette, and Tim Rocktäschel. 2020. Learning reasoning strategies in end-to-end differentiable proving. In *International Conference on Machine Learning*.

Farhad Moghimifar, Lizhen Qu, Yue Zhuo, Mahsa Baktashmotlagh, and Gholamreza Haffari. 2020. Cosmo: Conditional seq2seq-based mixture model for zero-shot commonsense question answering. In *Proceedings of the 28th International Conference on Computational Linguistics*.

Hongyu Ren and Jure Leskovec. 2020. Beta embeddings for multi-hop logical reasoning in knowledge graphs. *Advances in Neural Information Processing Systems*.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Maarten Sap, Vered Shwartz, Antoine Bosselut, Yejin Choi, and Dan Roth. 2020. Introductory tutorial: Commonsense reasoning for natural language processing. *Association for Computational Linguistics (ACL 2020): Tutorial Abstracts*.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*.

Maria I Sessa. 2002. Approximate reasoning by similarity-based sld resolution. *Theoretical computer science*, 275(1-2):389–426.

Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. 2019. End-to-end structure-aware convolutional networks for knowledge base completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2018. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.