MCMH: Learning Multi-Chain Multi-Hop Rules
for Knowledge Graph Reasoning

Lu Zhang ♣ **Mo Yu** ♡ **Tian Gao** ♡ **Yue Yu** ♣

♣ Lehigh University   ♡ IBM Research

luz319@lehigh.edu   {yum, tgao}@us.ibm.com   yuy214@lehigh.edu

Abstract

Multi-hop reasoning approaches over knowledge graphs infer a missing relationship between entities with a multi-hop rule, which corresponds to a chain of relationships. We extend existing works to consider a generalized form of multi-hop rules, where each rule is a set of relation chains. To learn such generalized rules efficiently, we propose a two-step approach that first selects a small set of relation chains as a rule and then evaluates the confidence of the target relationship by jointly scoring the selected chains. A game-theoretical framework is proposed to this end to simultaneously optimize the rule selection and prediction steps. Empirical results show that our multi-chain multi-hop (MCMH) rules result in superior results compared to the standard single-chain approaches, justifying both our formulation of generalized rules and the effectiveness of the proposed learning framework.

1 Introduction

Knowledge graphs (KGs) represent knowledge of the world as relationships between entities, i.e., triples with the form *(subject, predicate, object)* (Bollacker et al., 2008; Suchanek et al., 2007; Vrandečić and Krötzsch, 2014; Auer et al., 2007; Carlson et al., 2010). Such knowledge resource provides clean and structured evidence for many downstream applications such as question answering. KGs are usually constructed by human experts, which is time-consuming and leads to highly incomplete graphs (Min et al., 2013). Therefore automatic KG completion (Nickel et al., 2011; Bordes et al., 2013; Yang et al., 2014; Chen et al., 2018; Socher et al., 2013; Lao et al., 2011) is proposed to infer a missing link of relationship r between a head entity h and a tail entity t.

Existing KG completion work mainly makes use of two types of information: 1) co-occurrence of
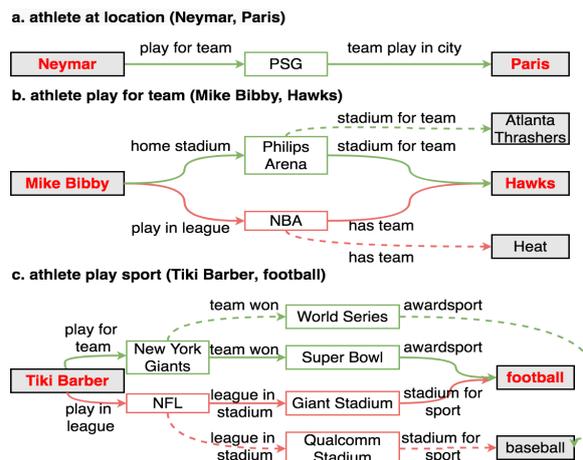
Figure 1: Examples of reasoning with multiple paths. (a) A standard multi-hop example. The target can be sufficiently inferred with one chain. (b) An example that requires a rule as the conjunction of two chains (the stadium hosts two teams but only one from NBA). (c) An example where multiple chains cannot sufficiently infer the target but improves its confidence.

entities and relations and 2) deducible reasoning paths of tuples. **KG embeddings** encode entities and relations, the first type of information, together into continuous vector space with low-rank tensor approximations (Bordes et al., 2013; Dettmers et al., 2017; Lin et al., 2015; Neelakantan et al., 2015; Shi and Weninger, 2017; Trouillon et al., 2016; Wang et al., 2014; Xie et al., 2016; Yang et al., 2014).

Ours approach utilizes the second type of information, reasoning path of tuples that can be deduced to the target tuple (Lao and Cohen, 2010; Xiong et al., 2017; Das et al., 2016, 2017). Here a reasoning path starts with the head entity h and ends with the tail entity t: $h \xrightarrow{r_1} e_1 \xrightarrow{r_k} e_k \xrightarrow{r_N} t$, where $r_1 \wedge ... \wedge r_N$ forms a relation chain that infers the existence of r. Therefore these methods are also referred as **multi-hop reasoning over KGs**, which learns a multi-hop chain as a rule to deduce the target r. An example of such a chain is given in

Figure 1a to infer whether an athlete plays in an location. Multi-hop reasoning approaches can usually utilize richer evidence and self-justifiable in terms of reasoning path rules used in the predictions, making the prediction of missing relations more interpretable.

Despite advantages and success of the multi-hop reasoning approach (Lin et al., 2018; Xiong et al., 2017; Das et al., 2017; Shen et al., 2018; Chen et al., 2018; Zhang et al., 2017), a target relationship may not be perfectly inferred from a single relation chain. There could exist multiple weak relation chains that correlate with the target relation. Figure 1 gives examples of such cases. These multiple chains could be leveraged in following ways: (1) the reasoning process naturally relies on the logic conjunction of multiple chains (Figure 1b); (2) more commonly, there are instances for which none of the chains is accurate, but aggregating multiple pieces of evidence improves the confidence (Figure 1c), as also observed in the case-based study works (Aamodt and Plaza, 1994; Das et al., 2020). Inspired by these observations, we propose the concept of **multi-chain multi-hop rule set**. Here, instead of treating each single multi-hop chain as a rule, we learn rules consisting of a small set of multi-hop chains. Therefore the inference of target relationships becomes a joint scoring of such a set of chains. We treat each set of chains as one rule and, since different query pairs can follow different rules, together we have a set of rules to reason each relation.

Learning the generalized multi-hop rule set is a combinatorial search problem. We address this challenge with a game-theoretic approach inspired by (Lei et al., 2016; Carton et al., 2018; Yu et al., 2019). Our approach consists of two steps: (1) selecting a generalized multi-hop rule set by employing a Multi-Layer Perceptron (MLP) over the candidate chains; (2) reasoning with the generalized rule set, which uses another MLP to model the conditional probability of the target relationship given the selected relation chains. The nonlinearity of MLP as reasoner provides the potential to model the logic conjunction among the selected chains in the rule set.

We demonstrate the advantage of our method on KG completion tasks in FB15K-237 and NELL-995. Our method outperforms existing single-chain approaches, showing that our defined generalized rules are necessary for many reasoning tasks.

2 Backgrounds

Problem Formulation We aim to infer missing relationships between two given entities, such as `athleteAtLocation` between `Neymar` and `Paris`, given their other connections in the knowledge graph. Formally, we are given a knowledge graph $\mathcal{G}$, consisting of a set of triplets $O = \{(h, r, t)\}$, where r is a relation edge defined in $\mathcal{G}$, h is a head entity, and t is the tail entity. The task is to identify the relation $\hat{r}$ between a set of query entity $\hat{h}$ and $\hat{t}$. For evaluation, we have ground truth labels indicating whether each pair $(\hat{h}, \hat{t})$ has the relationship $\hat{r}$ or not.

For a given query $(\hat{h}_i, \hat{r}, \hat{t}_i)$, the i-th sample in $\hat{r}$, we extract a set of relation chains $\mathcal{R} = \{\boldsymbol{R}_n\}_{n=1}^N = \{(\hat{h}, r_n^1, t_n^1), (t_n^1, r_n^2, t_n^2), \cdots (t_n^{m-1}, r_n^m, \hat{t})\}_{n=1}^N$ from the original KB $\mathcal{G}$. Each chain is a set of connected relations between $\hat{h}$ and $\hat{t}$ in $\mathcal{G}$. The proposed **multi-chain multi-hop rule set** is a set of rules, each consisting of multiple relation chains $\mathcal{S} \subset \mathcal{R}$ with size $d = |\mathcal{S}|$. In the experiments, we represent each relation chain $\boldsymbol{R}_n$ with only relation names. Our task is to find such $\mathcal{S}$ for a target relation $\hat{r}$ over each query pair $\hat{h}_i$ and $\hat{t}_i$, and estimate the confidence $P(\hat{r}|\mathcal{S})$. Note that $\mathcal{S}$ and $\mathcal{R}$ depend on query sample $(\hat{h}_i, \hat{r}, \hat{t}_i)$ but for notation simplicity we omit i and $\hat{r}$ from $\mathcal{S}_i^{\hat{r}}$ and $\mathcal{R}_i^{\hat{r}}$.

Relation Chains Extraction To obtain the set of candidate relation chains $\mathcal{R}$ for a target relation $\hat{r}$, we take the following extraction steps. First, we extract a fixed hop k sub-graph from the original KB. Each sub-graph starts with an entity $\hat{h}$ with relation $\hat{r}$, ends with an entity $\hat{t}$, and satisfies that $(\hat{h}, \hat{r}, \hat{t}) \in \mathcal{G}$. The sub-graph consists of a list of m-hop paths connecting the two ends, where $1 \leq m \leq k$. Each of the m-hop paths has the form $(\hat{h}, r^1, t^1), (t^1, r^2, t^2), \cdots (t^{m-1}, r^m, \hat{t})$. We call $r^1 \rightarrow r^2 \cdots \rightarrow r^m$ a candidate relation chain $\boldsymbol{R}$. High k values can result in an intractable number of chains while low k values may not have sufficient coverage. Here we extract chains with length up to $k = 3$, and for $\hat{r}$ with a large number of chains ($|\mathcal{R}| \geq 10^4$), we filter out extracted chains with a set threshold (proportional to count of relation chains) in the positive training data for that relation.

3 A Game-Theoretic Approach for MCMH Rule Learning

A Three-Player Game for Rule Learning Finding a set of chains as the rule is a combinatorial search problem in $\mathcal{R}$. For example, given an input
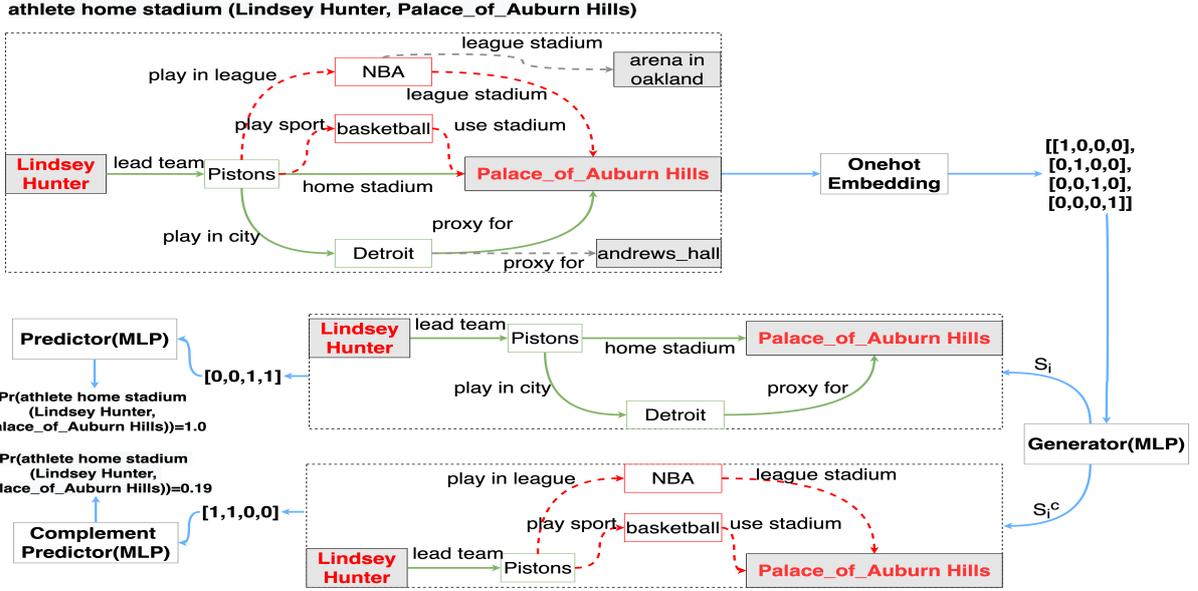
Figure 2: An example workflow of our model, with $|\mathcal{R}_i| = 4$. The generator selects the first two chains as the "critical information" for prediction: $\mathcal{S}_i = \{\texttt{LeadTeam} \to \texttt{HomeStadium}, \texttt{LeadTeam} \to \texttt{PlayinCity} \to \texttt{Proxyfor}\}$ with complement $\mathcal{S}_i^c = \{\texttt{LeadTeam} \to \texttt{PlayinLeague} \to \texttt{LeagueStadium}, \texttt{LeadTeam} \to \texttt{PlaySport} \to \texttt{UseStadium}\}$. In the prediction phase, the predictor $\mathcal{S}_i$ is encoded as $\mathbf{v}_{\mathcal{S}_i} = [0, 0, 1, 1]$ and estimates probability of $\texttt{athleteHomeStadium}$ being true as 100%. The complement predictor $\mathcal{S}_i^c$ is encoded as $\mathbf{v}_{\mathcal{S}_i^c} = [1, 1, 0, 0]$ and estimates the probablity as 19%.

of 1,000 chains between a training entity pair, the selection of a set-rule of 4 chains corresponds to a search space of 10^{12}. Hence, we propose a game-theoretic approximation to learn to generate predictive chains and reduce the learning complexity. Our method is inspired by the line of rationalization works (Carton et al., 2018; Yu et al., 2019). Specifically, our input is a set of chains $\mathcal{R}_i \subset \mathcal{R}$ for relation $\hat{r}$ and each training sample $(\hat{h}_i, \hat{r}, \hat{t}_i)$. Our method consists of three submodels: (1) a *rule set generator* that selects the set of chains $\mathcal{S}_i$ as a rule, (2) a *reasoner* that predicts the probability of $\hat{r}_i$ based on $\mathcal{S}_i$, and (3) a *complement predictor* that predicts the probability of $\hat{r}$ based on $\mathcal{S}_i^c = \mathcal{R}_i \setminus \mathcal{S}_i$.

During training, the *predictor* and the *complement predictor* aim to minimize the cross-entropy loss for predicting the existence of $\hat{r}$. While the *generator* is optimized to make the *predictor* perform well, while decreasing the *complement predictor*'s accuracy. In other words, the *generator* plays a cooperative game with the *predictor* to make the selected rule set $\mathcal{S}_i$ be useful for inferring the target relationship $\hat{r}$. At the same time it plays an adversarial game with the *complement predictor* to ensure that no critical information is left, i.e., to ensure the comprehensiveness of the selected $\mathcal{S}_i$. An example of the workflow is given in Figure 2.

Predictors The predictor estimates probability of $\hat{r}$ being true conditioned on $\mathcal{S}_i$, denoted as $\hat{p}(\hat{r}|\mathcal{S}_i)$.

The complement predictor estimates probability of $\hat{r}$ conditioned on $\mathcal{S}_i^c$, denoted as $\hat{p}^c(\hat{r}|\mathcal{S}_i^c)$. The two models are optimized as follows:

$$\mathcal{L}_p = \min_{\hat{p}} -H(p(\hat{r}|\mathcal{S}_i); \hat{p}(\hat{r}|\mathcal{S}_i)),$$
$$\mathcal{L}_c = \min_{\hat{p}^c} -H(p(\hat{r}|\mathcal{S}_i^c); \hat{p}^c(\hat{r}|\mathcal{S}_i^c)), \quad (1)$$

where $H(p; q)$ denotes the cross entropy between p and q, and $p(\cdot|\cdot)$ denotes the empirical distribution.

We encode the inputs $\mathcal{S}_i$ and $\mathcal{S}_i^c$ as binary vectors $\mathbf{v}_{\mathcal{S}_i}$ and $\mathbf{v}_{\mathcal{S}_i^c}$, respectively[1], which are both of dimension $|\mathcal{R}_i|$, with each dimension corresponding to one relation chain in the candidate set $\mathcal{R}_i$. The j-th component of $\mathbf{v}_{\mathcal{S}_i}$ is set to 1 if and only if the j-th chain is selected in $\mathcal{S}_i$, i.e., $\boldsymbol{R}_j \in \mathcal{S}_i$, and similarly for $\mathbf{v}_{\mathcal{S}_i^c}$. The input vectors are fed into a 3-layer MLP to predict whether $\hat{r}$ holds for $(\hat{h}_i, \hat{t}_i)$.

Generator The generator extracts $\mathcal{S}_i$ from the input chain set $\mathcal{R}_i$. This function, denoted as $g : \mathcal{R}_i \to \mathcal{S}_i$, is optimized with:

$$\min_{\boldsymbol{g}(\cdot)} \mathcal{L}_p - \mathcal{L}_c + \lambda_s \mathcal{L}_s, \quad (2)$$

where $\mathcal{L}_p$ and $\mathcal{L}_c$ are the losses of the *predictor* and the *complement predictor*, respectively. $\mathcal{L}_s$ is

---

[1]Our method could use KG embedding as inputs like previous works (Xiong et al., 2017; Das et al., 2017). It may weakens the interpretability of the reasoning model as they are smoothed representations, but can potentially improve the performance for cases with smaller training data. We leave the investigation to future work.

Dataset	#Entity	#Relation	#Triples	#Tasks
FB15K-237	14,505	237	310,116	10
NELL-995	75,492	200	154,213	10

Table 1: Statistics of the Datasets.

a sparsity loss which aims to constrain the number of chains to be select to a desired size d:

$$\mathcal{L}_s = \max\{(|\mathcal{S}_i| - d)/|\mathcal{R}_i|, 0\}. \tag{3}$$

Since the *generator* makes a hard decision for selection of $\mathcal{S}_i$, the losses $\mathcal{L}_p$ and $\mathcal{L}_c$ are generally not differentiable. Hence, we utilize the policy gradient (Williams, 1992) reinforcement learning algorithm to optimize the generator. To have bounded rewards, we use the predictors' accuracy instead of the loss values $\mathcal{L}_p$ and $\mathcal{L}_c$. The generator is also modeled with a MLP that is of the same architecture as the predictor. The output is a $|\mathcal{R}_i| \times 2$ vector which represents the probabilities that each chain would be selected into $\mathcal{S}_i$ and $\mathcal{S}_i^c$.

Rule selection during inference During inference, to have a fixed number (d) of selection, for each instance, we select the top-d chains according to the probability predicted by the generator.

4 Empirical Evaluation

We evaluate our model with MCMH rules on two datasets, FB15K-237 (Toutanova et al., 2015) and NELL-995 (Xiong et al., 2017). We follow the existing setting of treating each target relationship as a separate task and training and evaluating relationship-specific reasoning models, and use the standard data splits (Xiong et al., 2017). Table 1 summarizes statistics of two datasets. For each target relation in the datasets, we extract candidate chain set $\mathcal{R}$ following Section 2. Table 2 shows the number of extracted chains for each relation. We compare with previous works in the same setting, DeepPath (Xiong et al., 2017) and MINERVA (Das et al., 2017). They both are single-chain methods, i.e., they learn a reasoning model to find a single multi-hop chain for the inference.

Overall results Table 3 shows our method with double chains and five chains outperforms the single-chain baseline ($d = 1$ in our model) by clear margins on both datasets, demonstrating the advantage of our generalized rules compared to the single-chain rules studied in the existing works. Moreover, our generalized rule learning method,

when setting $d = 5$, outperforms existing baselines on both datasets. For some relations (such as the `teamSports` relation), our method performs worse compared to the previous works. It is likely because the relation has less training data while previous works use pre-trained KG embeddings to alleviate the problem.

Effects of numbers of chains in one rule (d) The required numbers of chains differ from different datasets: on NELL-995, using double-relation chain with $d = 2$ achieves slightly better performance compared to setting $d = 5$, while on FB15K-237 there is a clear advantage with $d = 5$ relation chains. This observation shows that on FB15K-237 a relation generally requires more chains as evidence to improve the confidence of prediction. Moreover, since a conjunction rule usually does not span over 5 chains, for many FB15K-237 test tuples the evidence is not sufficient for making the decision, therefore adding more chains can enhance the confidence thus improve results significantly.

Choices of d The average number of chains (i.e., the number of chains that connect the specific entity pair) is 13.8 for NELL-995 and 63.3 for FB15K-237. Therefore selecting d=5 chains is a significant portion of the whole input space. Moreover, MAP of our model using all candidate chains is 0.671 for FB15K-237 and 0.892 for NELL-995, which are close to that of d=5 (the detail performance for each relation is shown in Appendix B). From the above observations, selecting d=5 chains is sufficient for the KB completion task. Also, the logic conjunction between d=2 chains or among 5 chains is more likely to be human-interpretable compared to the selection of large numbers of chains. Figure 3 of Appendix B shows MAP versus the number of selected chains d for two representative relations, showing that the performance of our model converges after d=5.

Effects of MLP versus linear predictors Finally we study the impact of the two different ways that our generalized rules contribute to the improved results, namely *modeling logic conjunctions* and *enhancing confidence of multiple weak rules*, as discussed in Section 1. To this end, we replace the MLP predictors with linear models. The rationale is that the linear model is less effective in capturing conjunctions among inputs, so improvements from linear models over the single-chain baseline are more likely due to the enhanced confidence, rather than finding a conjunctive rule. We denote

FB15K-237			NELL-995		
Relation	**#Chains**	**#Chains per Sample**	**Relation**	**#Chains**	**#Chains per Sample**
teamSports	115	5.1	athletePlaysForTeam	852	20.9
birthPlace	285	62.5	athletePlaysInLeague	568	6.2
filmWrittenBy	153	65.9	athleteHomeStadium	174	5.2
filmDirector	132	37.5	athletePlaysSport	143	3.3
filmLanguage	3,380	82.2	orgHeadquaterCity	2,467	16.2
tvLanguage	1,614	55.2	orgHiredPerson	4,717	20.7
capitalOf	2,634	117.1	bornLocation	974	23.8
orgFounded	3,728	102.9	personLeadsOrg	3,347	20.3
musicianOrigin	6,784	158.2	teamPlaySports	228	6.3
personNationality	365	49.0	worksFor	4,840	21.6

Table 2: Number of chains extracted for each relation. We show both the total number of different chains for each relation, and the average number of chains that can be extracted per instance.

	Relation	**Single-Chain Baseline**	**Ours** $d=2$	**Ours** $d=5$	**Ours (-conj)** $d=2$	**Ours (-conj)** $d=5$	**DeepPath**	**MINERVA**
NELL-995	athletePlaysForTeam	0.872	0.940*	**0.947***	0.900	0.897	0.750	0.824
	athletePlaysInLeague	0.962	0.977*	**0.981***	0.957	0.975	0.960	0.970
	athleteHomeStadium	0.892	**0.896**	0.895	0.856	0.854	0.890	0.895
	athletePlaysSport	0.916	0.978*	0.982*	0.932	0.978	0.957	**0.985**
	teamPlaySports	0.728	0.769	0.782	0.669	0.771	0.738	**0.846**
	orgHeadquarterCity	0.957	0.932	0.907	**0.962**	0.903	0.790	0.946
	worksFor	0.794	0.842*	**0.849***	0.811	0.842	0.711	0.825
	bornLocation	0.823	**0.902***	0.850*	0.874	0.872	0.757	0.793
	personLeadsOrg	0.833	0.832	0.813	0.832	0.822	0.795	**0.851**
	orgHiredPerson	0.833	0.825	0.814	0.837	**0.855**	0.742	0.851
	Average	*0.861*	***0.890***	*0.882*	*0.863*	*0.877*	*0.809*	*0.879*
FB15K-237	teamSports	0.740	0.739	0.769*	0.758	0.765	**0.955**	-
	birthPlace	0.463	0.505*	**0.566***	0.443	0.512	0.531	-
	filmDirector	0.303	0.368	0.411*	0.363	0.413	**0.441**	-
	filmWrittenBy	0.498	0.516*	**0.553***	0.507	0.518	0.457	-
	filmLanguage	0.632	0.665*	**0.678***	0.667	0.675	0.670	-
	tvLanguage	**0.975**	0.962	0.957	0.957	0.956	0.969	-
	capitalOf	0.648	0.795	**0.825***	0.820	0.786	0.783	-
	orgFounded	0.465	0.407	**0.490***	0.431	0.485	0.309	-
	musicianOrigin	0.376	0.408*	**0.516***	0.390	0.476	0.514	-
	personNationality	0.713	0.806*	**0.828***	0.703	0.760	0.823	-
	Average	*0.581*	*0.617*	***0.659***	*0.604*	*0.635*	*0.645*	-

Table 3: Overall Results (MAP) on NELL-995 and FB15K-237. * highlights the cases where our MLP model outperforms the baseline with statistical significance (p-value<0.01 in t-test).

this model as **Ours (-conj)** and show its results in Table 3. It is observed that the **Ours (-conj)** model outperforms the baseline, but is generally not as good as the MLP model. Hence most of the relations mainly benefit from the case of confidence enhancement. However, the results also highlight a few relations with a notable performance gap, e.g., athletePlaysForTeam, indicating that multiple conjunctions are also important to KB completion tasks.

5 Conclusion

We propose a new approach of multi-chain multi-hop rule learning for knowledge graph completion tasks. First, we formalize the concept of multi-hop rule sets with multiple relation chains from knowledge graphs. Second, we propose a game-theoretical learning approach to efficiently select predictive relation chains for a query relation. Our formulation and learning method demonstrate advantages on two benchmark datasets over existing single-chain based approaches. For future work, we plan to investigate rules beyond chains, as well as integrate KG embeddings into our framework.

References

Agnar Aamodt and Enric Plaza. 1994. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, 7(1):39–59.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The Semantic Web*, pages 722–735, Berlin, Heidelberg. Springer Berlin Heidelberg.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data - SIGMOD08*. ACM Press.

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.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka, and Tom M. Mitchell. 2010. Toward an architecture for never-ending language learning. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI'10, page 1306–1313. AAAI Press.

Samuel Carton, Qiaozhu Mei, and Paul Resnick. 2018. Extractive adversarial networks: High-recall explanations for identifying personal attacks in social media posts. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3497–3507.

Wenhu Chen, Wenhan Xiong, Xifeng Yan, and William Yang Wang. 2018. Variational knowledge graph reasoning. *CoRR*, abs/1803.06581.

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, and Andrew McCallum. 2017. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning.

Rajarshi Das, Ameya Godbole, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. 2020. Nonparametric reasoning on knowledge bases. In *Automated Knowledge Base Construction (AKBC)*.

Rajarshi Das, Arvind Neelakantan, David Belanger, and Andrew McCallum. 2016. Chains of reasoning over entities, relations, and text using recurrent neural networks.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2017. Convolutional 2d knowledge graph embeddings.

Ni Lao and William W. Cohen. 2010. Relational retrieval using a combination of path-constrained random walks. *Machine Learning*, 81(1):53–67.

Ni Lao, Tom Mitchell, and William W Cohen. 2011. Random walk inference and learning in a large scale knowledge base. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 529–539. Association for Computational Linguistics.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117, Austin, Texas. Association for Computational Linguistics.

Xi Victoria Lin, Richard Socher, and Caiming Xiong. 2018. Multi-hop knowledge graph reasoning with reward shaping. *CoRR*, abs/1808.10568.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth AAAI conference on artificial intelligence*.

Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. 2013. Distant supervision for relation extraction with an incomplete knowledge base. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 777–782, Atlanta, Georgia. Association for Computational Linguistics.

Arvind Neelakantan, Benjamin Roth, and Andrew McCallum. 2015. Compositional vector space models for knowledge base completion. *CoRR*, abs/1504.06662.

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 International Conference on Machine Learning*, pages 809–816.

Yelong Shen, Jianshu Chen, Po-Sen Huang, Yuqing Guo, and Jianfeng Gao. 2018. Reinforcewalk: Learning to walk in graph with monte carlo tree search.

Baoxu Shi and Tim Weninger. 2017. Proje: Embedding projection for knowledge graph completion.

Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. *Advances in Neural Information Processing Systems*. 27th Annual Conference on Neural Information Processing Systems, NIPS 2013 ; Conference date: 05-12-2013 Through 10-12-2013.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 697–706, New York, NY, USA. ACM.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509, Lisbon, Portugal. Association for Computational Linguistics.

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

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata. *Communications of the ACM*, 57(10):78–85.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Twenty-Eighth AAAI conference on artificial intelligence*.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256.

Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. 2016. Representation learning of knowledge graphs with entity descriptions.

Wenhan Xiong, Thien Hoang, and William Yang Wang. 2017. Deeppath: A reinforcement learning method for knowledge graph reasoning.

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

Mo Yu, Shiyu Chang, Yang Zhang, and Tommi Jaakkola. 2019. Rethinking cooperative rationalization: Introspective extraction and complement control. 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 4094–4103, Hong Kong, China. Association for Computational Linguistics.

Ethan Zhang and Yi Zhang. 2009. *Average Precision*, pages 192–193. Springer US, Boston, MA.

Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander J. Smola, and Le Song. 2017. Variational reasoning for question answering with knowledge graph. *CoRR*, abs/1709.04071.