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)* [3, 22, 25, 2, 5]. Automatic KG completion [18, 4, 30, 7, 21, 13] is to infer a missing relationship r between a head entity h and a tail entity t. Existing KG completion work mainly utilizes two types of information: 1) co-occurrence of entities and relations; 2) deducible reasoning paths of tuples. **KG embeddings** encode entities and relations, the first type of information, into continuous vector space with low-rank tensor approximations [4, 11, 16, 17, 20, 24, 26, 28, 30]. Our approach utilizes the second type of information, reasoning path of tuples [12, 29, 8, 9]. Here a reasoning path starts from h and ends at 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. These methods are also referred as **multi-hop reasoning over KGs**, which learns a multi-hop chain as a rule to deduce the target. Multi-hop reasoning approaches can usually utilize richer evidence and are more interpretable.

Despite advantages of the multi-hop reasoning approach [15, 29, 9, 19, 7, 33], a target relationship may not be perfectly inferred from a single relation chain, and multiple chains could leverage the inference in two ways: (1) logic conjunction of multiple chains (Figure 1b); (2) aggregation of multiple pieces of evidence (Figure 1c), as also observed in the case-based study works [1, 10]. Hence, 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. Learning the generalized multi-hop rule set is a combinatorial search problem. Inspired by [14, 6, 31], we propose a game-theoretic approach consisting of two steps: (1) selecting a generalized multi-hop rule set with a Multi-Layer Perceptron (MLP); (2) using another MLP to model the conditional probability of the target relationship given the selected relation chains. We demonstrate the advantage of our method on KG completion tasks FB15K-237 and NELL-995. Our method outperforms existing single-chain approaches, showing that our defined generalized rules are necessary for many reasoning tasks.

4th Knowledge Representation and Reasoning Meets Machine Learning Workshop (KR2ML 2020), at NeurIPS.

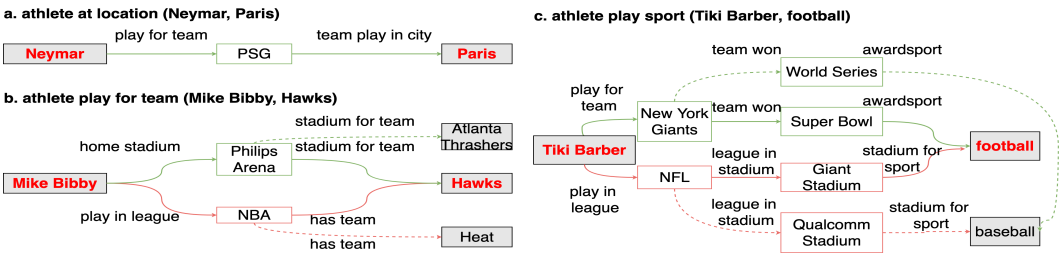

Figure 1: Examples of reasoning with multiple paths. Solid lines are selected paths (different colors indicate different paths), and dotted lines are unselected paths.

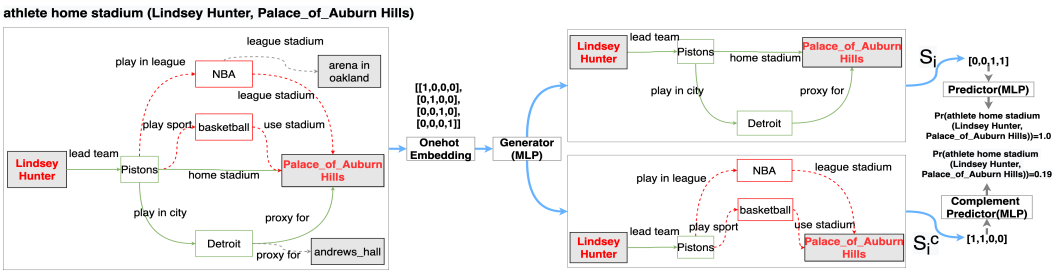

Figure 2: An example workflow of our model. The generator selects the green chains as the "critical information" for prediction. The predictor $\mathcal{S}_i$ is encoded as $\mathbf{v}_{\mathcal{S}_i}$=$[0, 0, 1, 1]$ and estimates probability of $\hat{r}$ being true as 100%. The complement predictor $\mathcal{S}_i^c$ is encoded as $\mathbf{v}_{\mathcal{S}_i^c}$=$[1, 1, 0, 0]$ and estimates the probablity as 19%.

2 Backgrounds

Problem Formulation We aim to infer missing relationships. Formally, we are given a KB $\mathcal{G}$, consisting of a set of triplets $O = \{(h, r, t)\}$, where r is a relation edge 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 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 $d = |\mathcal{S}|$. 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 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 $(\hat{h}, \hat{r}, \hat{t}) \in \mathcal{G}$. 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 $\boldsymbol{R}$. Here we extract chains with length up to $k = 3$.

3 A Game-Theoretic Approach for MCMH Rule Learning

A Three-Player Game for Rule Learning We propose a game-theoretic approximation to learn to generate predictive chains. The 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 red$\hat{r}$ 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. An example of the workflow is given in Figure 2.

Predictors The predictor estimates probability of $\hat{r}$ 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

FB15K-237			NELL-995		
Relation	**#Chains**	**#Chains/Sample**	**Relation**	**#Chains**	**#Chains/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 1: 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.

models are optimized as follows:

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

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:

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

where $\mathcal{L}_p$ and $\mathcal{L}_c$ are the losses of the *predictor* and the *complement predictor*, respectively. $\mathcal{L}_s$ is a sparsity loss which aims to constrain the number of chains to be select to a desired size d:

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

We utilize the policy gradient [27] 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.

4 Empirical Evaluation

We evaluate our model with MCMH rules on two datasets, FB15K-237 [23] and NELL-995 [29]. 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 [29]. The statistics of datasets are provided in Table 3, Appendix A. For each target relation in the datasets, we extract candidate chain set $\mathcal{R}$ following Section 2. Table 1 shows the number of extracted chains for each relation. We compare with previous single-chain works in the same setting, DeepPath [29] and MINERVA [9]. They both learn a reasoning model to find a single multi-hop chain for the inference.

Overall results Table 2 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. Moreover, our generalized rule learning method, when setting d=5, outperforms existing baselines on both datasets. For some relations (e.g.,the `teamSports` relation), our approach performs worse than previous works. It is likely caused by 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

[1] Our method could use KG embedding as inputs like previous works [29, 9]. 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.

	Relation	Single-Chain Baseline	Ours		Ours (-conj)		DeepPath	MINERVA
			d=2	d=5	d=2	d=5		
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 2: 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).

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). 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*. 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 this model as **Ours (-conj)** and show its results in Table 2. It is observed that most of the relations mainly benefit from the case of confidence enhancement, while multiple conjunctions are also important for some relations.

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.

Acknowledgments

L. Zhang and Y. Yu are supported by the National Science Foundation under award DMS 1753031.

References

[1] Agnar Aamodt and Enric Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, 7(1):39–59, March 1994. ISSN 0921-7126. URL http://dl.acm.org/citation.cfm?id=196108.196115.

[2] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In Karl Aberer, Key-Sun Choi, Natasha Noy, Dean Allemang, Kyung-Il Lee, Lyndon Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux, editors, *The Semantic Web*, pages 722–735, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-76298-0.

[3] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data - SIGMOD08*. ACM Press, 2008. doi: 10.1145/1376616.1376746. URL https://doi.org/10.1145/1376616.1376746.

[4] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795, 2013.

[5] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka, and Tom M. Mitchell. 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, 2010.

[6] Samuel Carton, Qiaozhu Mei, and Paul Resnick. 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, 2018.

[7] Wenhu Chen, Wenhan Xiong, Xifeng Yan, and William Yang Wang. Variational knowledge graph reasoning. *CoRR*, abs/1803.06581, 2018. URL http://arxiv.org/abs/1803.06581.

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

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

[10] Rajarshi Das, Ameya Godbole, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. Non-parametric reasoning on knowledge bases. In *Automated Knowledge Base Construction (AKBC)*, 2020.

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

[12] Ni Lao and William W. Cohen. Relational retrieval using a combination of path-constrained random walks. *Machine Learning*, 81(1):53–67, July 2010. doi: 10.1007/s10994-010-5205-8. URL https://doi.org/10.1007/s10994-010-5205-8.

[13] Ni Lao, Tom Mitchell, and William W Cohen. 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, 2011.

[14] Tao Lei, Regina Barzilay, and Tommi Jaakkola. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1011. URL https://www.aclweb.org/anthology/D16-1011.

[15] Xi Victoria Lin, Richard Socher, and Caiming Xiong. Multi-hop knowledge graph reasoning with reward shaping. *CoRR*, abs/1808.10568, 2018. URL http://arxiv.org/abs/1808.10568.

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

[17] Arvind Neelakantan, Benjamin Roth, and Andrew McCallum. Compositional vector space models for knowledge base completion. *CoRR*, abs/1504.06662, 2015. URL http://arxiv.org/abs/1504.06662.

[18] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 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, 2011.

[19] Yelong Shen, Jianshu Chen, Po-Sen Huang, Yuqing Guo, and Jianfeng Gao. Reinforcewalk: Learning to walk in graph with monte carlo tree search, 2018. URL https://openreview.net/forum?id=HJCRT81DM.

[20] Baoxu Shi and Tim Weninger. Proje: Embedding projection for knowledge graph completion, 2017. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14279/13906.

[21] Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. Reasoning with neural tensor networks for knowledge base completion. *Advances in Neural Information Processing Systems*, January 2013. ISSN 1049-5258. 27th Annual Conference on Neural Information Processing Systems, NIPS 2013 ; Conference date: 05-12-2013 Through 10-12-2013.

[22] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 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, 2007. ACM. ISBN 978-1-59593-654-7. doi: 10.1145/1242572.1242667. URL http://doi.acm.org/10.1145/1242572.1242667.

[23] Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 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, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1174. URL https://www.aclweb.org/anthology/D15-1174.

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

[25] Denny Vrandečić and Markus Krötzsch. Wikidata. *Communications of the ACM*, 57(10):78–85, September 2014. doi: 10.1145/2629489. URL https://doi.org/10.1145/2629489.

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

[27] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992. URL http://www.cs.ualberta.ca/~sutton/williams-92.pdf.

[28] Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. Representation learning of knowledge graphs with entity descriptions, 2016. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12216.

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

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

[31] Mo Yu, Shiyu Chang, Yang Zhang, and Tommi Jaakkola. 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, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1420. URL https://www.aclweb.org/anthology/D19-1420.

[32] Ethan Zhang and Yi Zhang. *Average Precision*, pages 192–193. Springer US, Boston, MA, 2009. ISBN 978-0-387-39940-9. doi: 10.1007/978-0-387-39940-9_482. URL https://doi.org/10.1007/978-0-387-39940-9_482.

[33] Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander J. Smola, and Le Song. Variational reasoning for question answering with knowledge graph. *CoRR*, abs/1709.04071, 2017. URL http://arxiv.org/abs/1709.04071.

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

Table 3: Statistics of the Datasets.

A Hyper-parameters and Reproducibility Checklist

Implementation dependencies libraries Preprocess: networkx 2.4. Model: Pytorch 1.4.0, cuda10.2.

Computing infrastructure The experiments run on servers with Intel(R) Xeon(R) CPU E5-2650 v4 and Nvidia GPUs (can be one of Tesla P100, V100, GTX 1070,or K80). The allocated RAM is 150G. GPU memory is 8G.

Data preprocess The statistics of original datasets are shown in Table 3. For the training set, we do the downsampling on the negative samples. We split the training and dev sets with the ratio of 0.8.

Model description There are 3 parts in our model, predictor, complement predictor, generator.

- MLP: Each part employs 3 linear layers with ReLU as activation. The dimension of each layer is half of that in the previous layer.
- Linear: The generator has the same structure as MLP, but the predictor and complement predictor have only one linear layer.

Average runtime for each approach

- MLP: The training time varies for each task, ranging from 8 hours to 50 hours. The main factor in the time variance is the size of the combinatorial action space.
- Linear: Training time ranges from 4 hours to 30 hours.

Number of model parameters The trainable parameter number of our model is task-specific, because the rules number varies for different relation tasks. For a task with D numbers of rules, the number of parameters of our MLP model is:

$$g(D) = 3\left(D \times \frac{D}{2} + \frac{D}{2} \times \frac{D}{4} + \frac{D}{4} \times 2\right) = \frac{15D^2}{8} + \frac{3D}{2}.$$

For instance, in the task of `personNationality`, D = 365. The number of parameters for each model is:

- MLP: 250,347
- Linear: 84,913

Corresponding validation performance for each reported test result The validation results of NELL-995 are listed in Table 4.

Explanation of evaluation metrics used In our experiment, we use Mean Average Precision (MAP) [32] as the evaluation metric.

Hyper-parameters We do not conduct extensive hyper-parameter tuning. In all tests we set learning rate of Adam as 0.001 and batch size as 20. Embedding dimension is the number of rules for each relation task. The weight for sparsity loss is set as $\lambda_s = 1.0$.

	Relation	Single-Chain Baseline	Ours		Ours (-conj)		DeepPath	MINERVA
			d=2	d=5	d=2	d=5		
NELL-995	athletePlaysForTeam	0.946	0.964	0.962	0.954	0.955	0.750	0.824
	athletePlaysInLeague	0.963	0.965	0.971	0.955	0.967	0.960	0.970
	athleteHomeStadium	0.918	0.931	0.945	0.936	0.922	0.890	0.895
	athletePlaysSport	0.942	0.955	0.960	0.934	0.959	0.957	0.985
	teamPlaySports	0.837	0.830	0.825	0.771	0.830	0.738	0.846
	orgHeadquarterCity	0.963	0.961	0.959	0.944	0.916	0.709	0.946
	worksFor	0.902	0.953	0.913	0.938	0.913	0.711	0.825
	bornLocation	0.955	0.939	0.950	0.930	0.946	0.757	0.793
	personLeadsOrg	0.984	0.966	0.981	0.9571	0.983	0.795	0.851
	orgHiredPerson	0.893	0.890	0.886	0.881	0.867	0.742	0.851
	average	0.930	0.935	0.935	0.920	0.926	0.809	0.879

Table 4: Overall results (MAP) on validation set of NELL-995.

B Results with All Chains

The idea in our paper is reasoning with more than one chains could improve KB completion performance, since they contain more information. So we perform experiments with d=all and show the results in Table 5. In these experiments there is no generator. All chains between the given query $(\hat{h}, \hat{r}, \hat{t})$ are taken as the input of the predictor. From intuition, with more evidence a higher MAP is generally expected. We therefore use these results as a reference upperbound of our method. [2]

FB15K-237		NELL-995	
Relation	d=**all**	**Relation**	d=**all**
teamSports	0.791	athletePlaysForTeam	0.946
birthPlace	0.577	athletePlaysInLeague	0.970
filmWrittenBy	0.579	athleteHomeStadium	0.864
filmDirector	0.420	athletePlaysSport	0.977
filmLanguage	0.696	orgHeadquaterCity	0.935
tvLanguage	0.960	orgHiredPerson	0.851
capitalOf	0.817	bornLocation	0.828
orgFounded	0.508	personLeadsOrg	0.836
musicianOrigin	0.527	teamPlaySports	0.839
personNationality	0.834	worksFor	0.869
Average	0.671	Average	0.892

Table 5: MAP Results of our predictor with all chains (d=∞).

C Additional Experiments on Top-K Generation from the Single-Chain Baseline

We add an additional experiment, *Single-Chain Gen*, as an additional baseline in this part. Since we train the generator and predictor together at the same time in our method, we are interested in the performance of the predictor without knowing the target d (i.e., the number of selected chains). In this experiment, we first train a singe-chain model to obtain a generator, then take the top d=2 or 5 chains from the resultant generator and train the predictor separately. From the results shown in Table 6, it can be observed that our proposed model also outperforms this new baseline. Hence our model does capture the conjunction information among the chains during the subset selection procedure in the generator phase.

[2] Precisely, this result could not show the real upperbound of reasoning task with more than one chains. This is due to (1) the capacity of the MLP models may not be sufficient to capture the conjunction among all chains; (2) the reported numbers are affected by the generalizability of models and randomness of the data.

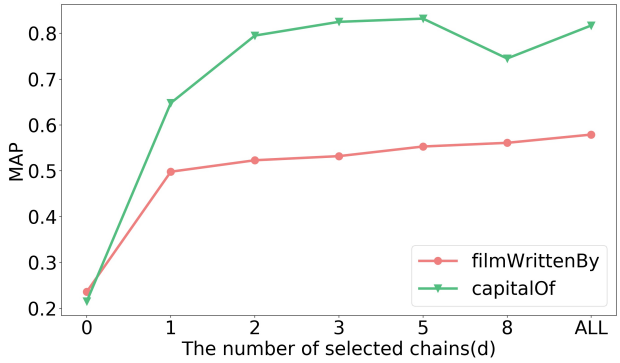

Figure 3: MAP with d increasing.

	Relation	Single-Chain Baseline	Singe-Chain Gen d=2	Singe-Chain Gen d=5	Ours d=2	Ours d=5	DeepPath	MINERVA
NELL-995	athletePlaysForTeam	0.872	0.898	0.913	0.940	**0.947**	0.750	0.824
	athletePlaysInLeague	0.962	0.957	0.977	0.977	**0.983**	0.960	0.970
	athleteHomeStadium	0.892	0.859	0.856	**0.896**	0.895	0.890	0.895
	athletePlaysSport	0.916	0.911	0.978	0.978	0.982	0.957	**0.985**
	teamPlaySports	0.728	0.690	0.775	0.769	0.782	0.738	**0.846**
	orgHeadquarterCity	**0.957**	0.955	0.953	0.932	0.907	0.790	0.946
	worksFor	0.794	0.859	0.850	0.842	**0.849**	0.711	0.825
	bornLocation	0.823	0.906	0.861	**0.902**	0.850	0.757	0.793
	personLeadsOrg	0.833	0.817	0.784	0.832	0.813	0.795	**0.851**
	orgHiredPerson	0.833	0.833	**0.852**	0.825	0.814	0.742	0.851
	Average	*0.861*	*0.868*	*0.880*	***0.889***	*0.882*	*0.809*	*0.879*
FB15K-237	teamSports	0.740	0.746	0.743	0.739	0.769	**0.955**	-
	birthPlace	0.463	0.517	0.512	0.505	**0.566**	0.531	-
	filmDirector	0.303	0.271	0.272	0.368	0.411	**0.441**	-
	filmWrittenBy	0.498	0.523	0.544	0.516	**0.553**	0.457	-
	filmLanguage	0.632	0.687	0.684	0.665	**0.678**	0.670	-
	tvLanguage	**0.975**	0.967	0.968	0.962	0.957	0.969	-
	capitalOf	0.648	0.740	0.758	0.795	**0.825**	0.783	-
	orgFounded	0.465	0.441	0.472	0.407	**0.490**	0.309	-
	musicianOrigin	0.376	0.419	0.468	0.408	**0.516**	0.514	-
	personNationality	0.713	0.813	0.825	0.806	**0.828**	0.823	-
	Average	*0.581*	*0.612*	*0.625*	*0.617*	***0.659***	*0.645*	-

Table 6: Overall Results (MAP) on NELL-995 and FB15K-237 single chain generator and MLP predictor.