Path-based Link Prediction on Hyper-relational Knowledge Graph

Shuzhi Liu
Department of Electronic Engineering
Tsinghua University
Beijing, China
liushuzhi@mail.tsinghua.edu.cn

Shimin Di
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
Hong Kong SAR, China
sdiaa@connect.ust.hk

Jianwen Peng
Department of Electronic Engineering
Tsinghua University
Beijing, China
jianwenpeng02@outlook.com

Quanming Yao
Department of Electronic Engineering
Tsinghua University
Beijing, China
qyaoaa@mail.tsinghua.edu.cn

***Abstract*—Recently, path-based Graph Neural Networks (GNNs) have achieved promising performance in the link prediction task on benchmark Knowledge Graphs (KGs). However, there is no research on leveraging path-based GNNs to promote the more general case of KGs, namely hyper-relational KGs (HKGs). To bridge this research gap, we study the path-based GNNs and discover that existing path-based GNNs fail to handle HKGs because they cannot well explore the external information (i.e., qualifiers) stored in HKGs. In this paper, we propose a novel framework, Hyper Path-based Graph Neural Network (HyperPGNN) for HKGs. Specifically, we propose a novel Hyper2Tri conversion and hyper query learner to better enable the path-based GNNs to understand qualifiers in HKG, and then capture them into the graph learning. Results show our method achieves good performance in both transductive and inductive settings. Codes are available at https://github.com/LARS-research/HyperPGNN.**

***Index Terms*—Knowledge graph, link prediction, hyper-relational graph, graph neural network**

I. Introduction

Hyper-relational Knowledge Graphs (HKGs) allow equipping the main triplet (h, r, t) with qualifiers $\{(qr_k : qe_k)\}$, providing contextual information for (h, r, t). As the example in Fig. 1, the fact *(Albert Einstein, employer, University of Zurich)* with qualifiers $\{$*(start time: 1909), (end time: 1911)*$\}$ can be distinguished from the fact *(Albert Einstein, employer, Swiss Federal Institute of Intellectual Property)* with qualifiers $\{$*(start time: 1902), (end: 1909)*$\}$. Like KG, link prediction is a fundamental task in HKGs that predicts missing facts based on known ones. Recently, a series of methods extends the classic KG embedding models from KGs to HKGs, e.g., m-TransH [1] from TransH [2], GETD [3] from TuckER [4], StarE [5] from CompGCN [6].

Despite the success of existing methods for HKGs, those powerful path-based graph neural networks (GNNs) [7]–[9] that achieved outstanding performance on benchmark KGs have not been explored on the more general HKGs. Different from aggregating the information from neighbors [6], [10], path-based GNNs leverage the information from the possible paths between head and tail entities to predict the link. Inspired by their success, the path-based idea may also work on HKGs. For example, from a path $a \xrightarrow{\text{employer}} b \xrightarrow{\text{location}} c$ together with qualifiers $\{$*(start time: 1909), (end time: 1911)*$\}$ as constraints of the *employer* relation, we can infer that $\{(a, \text{work location}, c), (\text{start time} : 1909), (\text{end time} : 1911)\}$. The a, b and c here can be any other entities and do not have to be the same as those in Fig. 1,

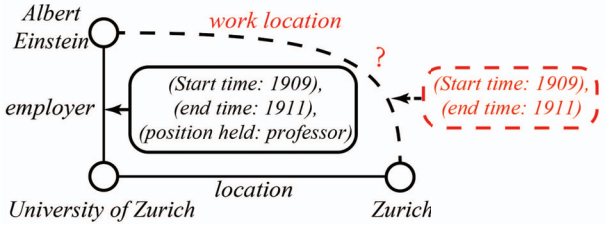

Fig. 1. Illustration of hyper-relational facts in HKGs.

However, it is a non-trivial task to extend existing path-based GNNs to HKGs. Current path-based works can only traverse and leverage the main triplets without qualifiers in HKGs. Thus, they will miss the important information stored in the qualifiers, like the working period in Figure 1. Therefore, to leverage the capability of path-based GNNs on HKGs, we propose a new model named **Hyper P**ath-based **G**raph **N**eural **N**etwork (HyperPGNN) in this paper. More concretely, we propose a conversion method Hyper2Tri to covert the topology of the hyper-relational facts into the structures that can be incorporated with existing path-based GNNs. After the topology conversion, we design a query learner to encode qualifiers to further capture the information

This work is supported by National Natural Science Foundation of China (No. 92270106)

979-8-3503-5409-6/24/$31.00 ©2024 IEEE
DOI 10.1109/CAI59869.2024.00194

of qualifiers. Empirical results show that HyperPGNN achieves good performance on benchmark HKG datasets under both transductive and inductive settings.

II. Preliminary

A. Link prediction in HKGs

A HKG can be defined as $\mathcal{G} = (\mathcal{V}, \mathcal{R}, \mathcal{D})$, where $\mathcal{V}$ is the set of entities, $\mathcal{R}$ represents the set of relations and $\mathcal{D}$ contains a set of hyper-relational facts. Each hyper-relational fact consists of a main triplet (h, r, t) and a list of qualifiers $\{(qr_k : qe_k)\}_{k=1}^{n}$, where $h, t, qe_k \in \mathcal{V}$ and $r, qr_k \in \mathcal{R}$. Link prediction on HKGs aims to predict missing head or tail entities in the query hyper-relational fact $\{(?, r, t), (qr_k : qe_k)_{k=1}^{n})\}$ or $\{(h, r, ?), (qr_k : qe_k)_{k=1}^{n})\}$. For simplicity, we use the notation of the tail prediction in the following part.

B. Path-based link prediction framework for KGs

For KGs, path-based link prediction framework infers relationships between two entities h and t through the paths connecting them. For example, from a path $a \xrightarrow{\text{employer}} b \xrightarrow{\text{location}} c$ we can infer that $(a, worklocation, c)$. Note that the entities are not considered and only the relations are used to compute the representation of entity pair (h, t).

C. Path-based Graph Neural Networks for KGs

The path-based GNN on KGs initializes the head entity's representation with a special indicator vector, while other entities are initialized with zero vectors. The indicator vector is typically selected to be the embedding of the query relation to make the final output query-dependent. Messages only depend on edge type and query relation embedding, unrelated to entities. After message passing, the embedding at the tail entity is read out and fed to a MLP to compute the probability of query relation existence between them. With these design of embedding initialization and message passing, only the relations in the paths contribute to the final output, so these GNNs can be considered as path-based frameworks.

Without using entity embeddings, path-based GNNs naturally generalize to the inductive setting. They are proved to be more expressive than the neighbor-based GNNs theoretically [11], and have better performance empirically [7]–[9]. However, without mechanisms to deal with qualifiers, existing path-based GNNs cannot be directly applied to HKGs.

III. Proposed Method

We propose HyperPGNN to extend the path-based GNNs to HKGs. The overall framework of the model is shown in Figure 2. The challenges in designing a path-based neural networks for HKGs are two-fold: (i) For the known facts, expressing the information present in the qualifiers. (ii) For the query facts, incorporating the qualifier constraints when conducting link predictions. For the challenge (i), we propose a Hyper2Tri conversion to utilize the qualifier information in known facts (section III-A). For the challenge (ii), we design a hyper query learner to incorporate the qualifiers in the query (section III-B). Then, we adopt an advanced path-based GNN to get an expressive representation of possible hyper-relational facts (section III-C).

A. Hyper Facts Conversion

Intuitively, we may convert the qualifiers into collections of triplets to facilitate path-based embedding with the restricting information from statement qualifiers. Existing conversion approaches, e.g., Star-to-Clique proposed by m-TransH [1], generally treat qualifier entities equally with triplet entities. They first reformulate the statement as a set of relation-entity pairs $P = (\{r_h : h\}, \{r_t : t\}, \{qr_i : qe_i\}_{i=1}^{n})$, then form a triplet $(e_1, r_1\text{-}r_2, e_2)$, for every $\{r_1 : e_1\}, \{r_2 : e_2\} \in P$. It has two major defects: 1) losing the semantic difference between triplet entities and qualifier entities, 2) introducing too much noise while expanding possible path formulation.

Thus, we propose the following Hyper2Tri (Hyper-relation to Triangle relations) conversion. Given a statement $\{(h, r, t), (qr_k : qe_k)_{k=1}^{n})\}$, every $(qr_k : qe_k)$ is converted to two labeled edges $(h, r\text{-}qr_k, qe_k)$ and $(t, r\text{-}qr_k, qe_k)$. Here, $r\text{-}qr_k$ is the new type of relation representing qualifier relation qr_k of relation r, and we denote its type as the pair of r and qr_k. See the Hyper Facts Conversion part of Figure 2 for an illustration. In other words, a statement with n qualifiers will be decomposed into $2n + 1$ triplets, including the main triplet and $2n$ triplets derived from the qualifiers. Hence, the original HKG is transformed into a knowledge graph that can be processed by previous path-based link prediction frameworks without losing information in qualifiers. Note that the newly constructed triplets will form new paths together with existing main triplets, providing more information about the qualifiers. As shown in Figure 3, the proposed conversion produced new paths related to the *Argentina national association football team*, making it possible to infer that the nationality of *Lionel Messi* is *Argentina*, and he won the *2022 FIFA World Cup*.

Compared with Star-to-Clique [1] Hyper2Tri strengthens the connection between the primal entity and qualifier entities while weakening the bonds among qualifier entities. And Hyper2Tri creates no new bonds between qualifier entities, thus reducing information noise. Besides, another possible design is to directly encode the hyper-relations in the paths and using these representations to conduct predictions. However, the performance will be limited without collecting path-based information about qualifiers.

B. Hyper Query Learner

We design a hyper query learner to achieve the embedding $\mathbf{q}$ of a given query $\{(h, r, ?), (qr_k : qe_k)_{k=1}^{n})\}$ [1]:

$$\mathbf{q} \leftarrow \gamma(\mathbf{r}, \mathbf{W}_r \cdot \sum_k \phi(\mathbf{qr}_k, \mathbf{qe}_k)), \tag{1}$$

where $\mathbf{qr}_k, \mathbf{qe}_k$ and $\mathbf{r}$ are the learned embedding of qr_k, qe_k and r respectively. $\phi(\cdot)$ is the function to composite the embedding of qualifier keys and qualifier values, and can be any function of the form $\mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n$. We sum the

[1]We use the same notation for the query facts as known facts here to maintain symbol simplicity.

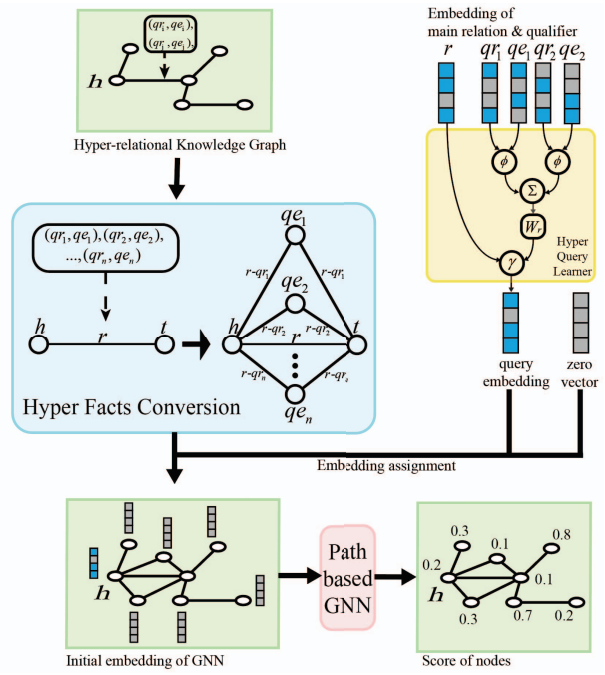

Fig. 2. An overview of our pipeline

composition results of all qualifiers to get the embedding of qualifiers, and then transform with a relation-specific matrix W_r. $\gamma(\cdot)$ is another composition function.

In this way, we integrate the information from the qualifiers included in the query into the embeddings, enabling the final prediction results to better satisfy the constraints imposed by the qualifiers. Note that the above process does not need the embedding of the head entity h , which is crucial for applying to the inductive setting.

C. Path-based GNN

Given a HKG $\mathcal{G}$, we convert it to a knowledge graph $\tilde{\mathcal{G}}$ as described in Section III-A. For a query $\{(h, r, ?), (qr_k : qe_k)_{k=1}^n)\}$, we initialize the representation on entity h with the hyper query embedding $\mathbf{q}$ obtained in Equation 1. The representations of other entities are initialized with zero vectors. Then the initial representations are fed into a Path-based GNN. The two modules proposed above can be integrated with any path-based methods designed for KGs. We adopt the NBFNet [7] to compute the final scores of all candidates.

IV. Experiment

A. Dataset

Experiments are conducted under both transductive and inductive settings. For the transductive setting, we use Wikipeople [12], JF17K [13], and WD50K [5]. For the inductive setting, we use WP-IND, JF-IND, and MFB-IND [14]. In a transductive setting during testing, all the entities and relations in the main triplet and qualifiers are limited to those that have occurred in the training set. However, in the inductive setting,

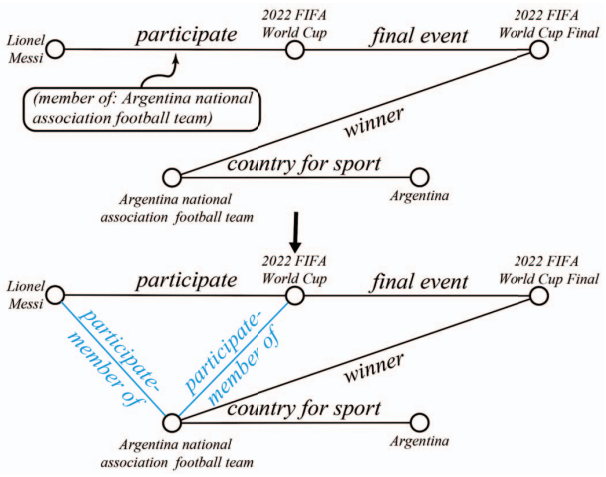

Fig. 3. Hyper2Tri will form new paths for qualifiers

the model is required to predict query facts that may involve new head entities and new tail entities as potential candidates.

B. Experimental Setup

a) Base models: For the transductive setting, We compare the proposed method with StarE [5], Hy-Transformer [15], GRAN [16] and QUAD [17]. For the inductive setting, we compare our results with GMPNN [14].

b) Metric: We report the mean reciprocal ranking(MRR) and Hit@1,10 for the transductive setting. For inductive setting, we report MRR and Hit@1,3 to keep in line with existing work. The reported results are averaged over five runs.

c) Network Details: For the hyper query learner, we use the rotate function [18] as the $\phi(\cdot)$ in Equation 1 and $\gamma(\cdot)$ is a weighted sum function with the weight of qualifiers is set to 0.4. We set the embedding size to be 32. For the path-based GNN, we use a GNN with 6 layers. We use the DisMult function proposed by DisMult [19] as the message function and the pna [20] for aggregation. A two-layer MLP with a hidden layer of size 32 is used to compute the final scores.

d) Training: We train the model in 1-N setting with binary cross entropy loss. We employ Adam to train the model for at most 1000 epochs, the learning rate is 0.001. We use a NVIDIA GTX 3090 GPU for experiments.

C. Performance Comparison

Table I and Table II show the performance comparison in the transductive setting and inductive setting respectively. The results of the baselines were gathered from the original literature, and "-" indicates the results were not reported. When comparing with G-MPNN, the original model also predicts of values in qualifiers, we retrain the model to only predict head and tail entities and get slightly better results. It can be seen that our model consistently outperforms the state-of-the-art.

D. Ablation Study

We compare **FullModel** with the following variants. (i) **BaseModel** eliminates the Hyper2Tri, dropping qualifiers in

TABLE I
COMPARISON ON TRANSDUCTIVE DATASETS. THE BEST RESULTS ARE IN BOLD. THE SECOND-BEST RESULTS ARE UNDERLINED

Method	WD50K			WikiPeople			JF17K		
	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10
m-TransH	-	-	-	0.063	0.063	0.300	0.206	0.206	0.463
StarE	0.349	0.271	0.496	0.491	0.398	**0.648**	0.574	0.496	0.725
Hy-Transformer	0.356	0.281	0.498	0.501	0.426	0.634	0.582	0.501	0.742
GRAN-hete	-	-	-	**0.503**	0.438	0.620	0.617	0.539	0.770
QUAD	0.348	0.270	0.497	0.466	0.365	0.624	0.582	0.502	0.740
QUAD(Parallel)	0.349	0.275	0.489	0.497	**0.431**	0.617	0.596	0.519	0.751
HyperPGNN	**0.362**	**0.283**	**0.505**	0.501	0.430	**0.648**	**0.657**	**0.585**	**0.771**

TABLE II
COMPARISON ON INDUCTIVE DATASETS, THE BEST RESULTS ARE IN BOLD. THE SECOND-BEST RESULTS ARE UNDERLINED

Method	WP-IND			JF-IND			MFB-IND		
	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10
G-MPNN-sum	0.010	0.051	0.185	0.240	0.188	0.331	0.292	0.203	0.479
G-MPNN-mean	0.093	0.029	0.189	0.146	0.084	0.221	0.242	0.151	0.416
G-MPNN-max	0.160	0.093	0.293	0.224	0.159	0.315	0.268	0.191	0.283
HyperPGNN	**0.312**	**0.262**	**0.419**	**0.463**	**0.412**	**0.487**	**0.507**	**0.432**	**0.531**

TABLE III
ABATION STUDY IN TRANSDUCTIVE SETTING, HERE WE USE H@10 AS ABBREVIATION FOR HIT@10 TO SAVE SPACE

Method	WD50K		WP		JF17K	
	MRR	H@10	MRR	H@10	MRR	H@10
BaseModel	0.153	0.295	0.251	0.468	0.341	0.570
NoConversion	0.203	0.345	0.322	0.477	0.419	0.619
Star2Clique	0.297	0.425	0.416	0.508	0.543	0.748
NoQueryLeaner	0.276	0.462	0.383	0.471	0.597	0.763
FullModel	**0.362**	**0.505**	**0.499**	**0.648**	**0.657**	**0.771**

known facts, and removes the Hyper Query Learner, relying solely on main relation r for the query embedding. (ii) **NoConversion** removes the Hyper2Tri. (iii) **Star2Clique** substitute the Hyper2Tri with Star2Clique. (iv) **NoQueryLeaner** elimates Hyper Query Learner.

Table III presents the results of the variants in transductive setting. Comparing the FullModel, the NoConversion and the Star2Clique, we show the efficiency of the Hyper2Tri conversion. Comparing the FullModel and the No queryLearner, we show the importance of the hyper query learner. Comparing the Basemodel with Noconversion and NoQueryLearner, we show the two module contribute to performance independently.

V. CONCLUSION

We introduce a Hyper2Tri conversion and a hyper query learner to incorporate information of qualifiers into path-based link prediction model for HKGs. Tests on various datasets show superior performance in both inductive and transductive settings.

REFERENCES

[1] J. Wen, J. Li, Y. Mao, S. Chen, and R. Zhang, "On the representation and embedding of knowledge bases beyond binary relations," *arXiv preprint arXiv:1604.08642*, 2016.

[2] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 28, no. 1, 2014.

[3] Y. Liu, Q. Yao, and Y. Li, "Generalizing tensor decomposition for n-ary relational knowledge bases," in *Proceedings of the web conference 2020*, 2020, pp. 1104–1114.

[4] I. Balažević, C. Allen, and T. M. Hospedales, "Tucker: Tensor factorization for knowledge graph completion," *arXiv preprint arXiv:1901.09590*, 2019.

[5] M. Galkin, P. Trivedi, G. Maheshwari, R. Usbeck, and J. Lehmann, "Message passing for hyper-relational knowledge graphs," *arXiv preprint arXiv:2009.10847*, 2020.

[6] S. Vashishth, S. Sanyal, V. Nitin, and P. Talukdar, "Composition-based multi-relational graph convolutional networks," *arXiv preprint arXiv:1911.03082*, 2019.

[7] Z. Zhu, Z. Zhang, L.-P. Xhonneux, and J. Tang, "Neural bellman-ford networks: A general graph neural network framework for link prediction," *Advances in Neural Information Processing Systems*, vol. 34, pp. 29 476–29 490, 2021.

[8] Z. Zhu, X. Yuan, L.-P. Xhonneux, M. Zhang, M. Gazeau, and J. Tang, "Learning to efficiently propagate for reasoning on knowledge graphs," *arXiv preprint arXiv:2206.04798*, 2022.

[9] Y. Zhang, Z. Zhou, Q. Yao, X. Chu, and B. Han, "Adaprop: Learning adaptive propagation for graph neural network based knowledge graph reasoning," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 3446–3457.

[10] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *European Semantic Web Conference*, 2018, pp. 593–607.

[11] M. Zhang, P. Li, Y. Xia, K. Wang, and L. Jin, "Labeling trick: A theory of using graph neural networks for multi-node representation learning," *arXiv preprint arXiv:2010.16103*, 2020.

[12] S. Guan, X. Jin, Y. Wang, and X. Cheng, "Link prediction on n-ary relational data," in *Proceedings of the 28th International Conference on World Wide Web (WWW'19)*, 2019, pp. 583–593.

[13] P. Rosso, D. Yang, and P. Cudré-Mauroux, "Beyond triplets: hyper-relational knowledge graph embedding for link prediction," in *Proceedings of the web conference 2020*, 2020, pp. 1885–1896.

[14] N. Yadati, "Neural message passing for multi-relational ordered and recursive hypergraphs," in *Advances in Neural Information Processing Systems (NeurIPS) 33*. Curran Associates, Inc., 2020.

[15] D. Yu and Y. Yang, "Improving hyper-relational knowledge graph completion," *arXiv preprint arXiv:2104.08167*, 2021.

[16] Q. Wang, H. Wang, Y. Lyu, and Y. Zhu, "Link prediction on n-ary relational facts: A graph-based approach," *arXiv preprint arXiv:2105.08476*, 2021.

[17] H. Shomer, W. Jin, J. Li, Y. Ma, and J. Tang, "Learning representations for hyper-relational knowledge graphs," *arXiv preprint arXiv:2208.14322*, 2022.

[18] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "Rotate: Knowledge graph embedding by relational rotation in complex space," *arXiv preprint arXiv:1902.10197*, 2019.

[19] B. Yang, S. W.-t. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," in *Proceedings of the International Conference on Learning Representations (ICLR) 2015*, 2015.

[20] G. Corso, L. Cavalleri, D. Beaini, P. Liò, and P. Veličković, "Principal neighbourhood aggregation for graph nets," *Advances in Neural Information Processing Systems*, vol. 33, pp. 13 260–13 271, 2020.