Type-enhanced Inductive Knowledge Graph Completion

Suxue Ma[1], Zhe Wang[2], Kewen Wang[2,*] and Zhiqiang Zhuang[1]

[1]College of Intelligence and Computing, Tianjin University, Tianjin, China

[2]School of Information and Communication Technology, Griffith University, Brisbane, Australia

Abstract

Inductive knowledge graph completion has gained significant attention due to the dynamic nature of entities and facts in knowledge graphs (KGs). The goal of this task is to predict missing links between entities that are unseen during training. Graph neural networks (GNNs) have proven to be effective in handling this task. However, existing GNN-based methods overlook the type information of entities in KGs and thus may make incorrect predictions, which also limits the interpretability of the GNN-based models for KG completion. To address this limitation, we propose to incorporate type information into an existing GNN-based model for inductive KG completion. Experimental results show that our proposed approach is effective in improving the performance of inductive link prediction.

Keywords

Knowledge Graph, Link Prediction, Graph Neural Networks, Type Information

1. Introduction

Knowledge graphs (KGs) contain a vast amount of structured data, but they are often incomplete. Knowledge graph completion (KGC) is to predict missing links in KGs, which can be beneficial to downstream tasks such as recommender systems and question answering systems. While many models have been proposed for KGC, they are not effective for predicting the relations of entities that are unseen in the training. Inductive KGC aims to develop KGC models that are able to perform link prediction involving new entities. Graph neural networks (GNNs) have proven effective in handling this task [1][2]. However, existing GNN-based methods for inductive link prediction overlook the type information of entities in KGs and thus may make incorrect predictions. Utilizing type information may also enhance the interpretability of the GNN-based models for KG completion.

In fact, by transferring knowledge during training to the inference stage, type information facilitates inductive KGC in two ways. Firstly, GNN-based models make predictions solely based on subgraph structures, and incorporating type information as supplementary information can enhance performance. GNN-based models can predict a triple as true if the target entities (i.e., the head entity and the tail entity) are genuinely linked but through a relation different from the

ISWC 2023 Posters and Demos: 22nd International Semantic Web Conference, November 6–10, 2023, Athens, Greece

*Corresponding author.

msx@tju.edu.cn (S. Ma); zhe.wang@griffith.edu.au (Z. Wang); k.wang@griffith.edu.au (K. Wang); zhuang@tju.edu.cn (Z. Zhuang)

0000-0002-1367-7139 (Z. Wang); 0000-0002-0542-3761 (K. Wang); 0000-0003-0081-1703 (Z. Zhuang)

CEUR Workshop Proceedings (CEUR-WS.org)

target relation. For example, the target triple (albert_einstein, worksAt, theory_of_relativity) may be predicted as true because there is evidence in the subgraph suggesting albert_einstein and theory_of_relativity are connected. However, they are linked by the relation develops not worksAt. By imposing type constraints and recognizing that theory_of_relativity is not a workplace, we can readily rectify this classification error. Secondly, the inclusion of type information can enhance the interpretability of models. Without the guidance type constraints, models may generate predictions that are clearly wrong and can be easily recognized by humans, as exemplified by the erroneous prediction (albert_einstein, worksAt, theory_of_relativity). Introducing type information can help reduce such errors, thereby enhancing the model's reliability.

In this paper, we propose to incorporate type information from KGs into an existing GNN-based model for inductive knowledge graph completion. This is achieved by a novel integration of type information in subgraph structures by prioritizing triples that are more likely to adhere to the type constraints. This is not straightforward since type information can be incomplete and the numbers of entities of different types can be diverse. To resolve these issues, we develop a method for inferring new entity type information from the existing entities and type hierarchies. Our model is the first attempt for incorporating type information into inductive KGC models. We note that while several works have been done on utilising type information for standard KGC [3][4], they cannot be directly applied to inductive KGC. Just as most inductive KGC settings, we only consider unseen entities, while relations and type information are seen in the training set. This is because in real-life applications, the relations and type information are usually more stable whereas entities may change. For instance, in e-commerce platforms, new products and users continually emerge, yet their types often remain consistent with the existing knowledge graph. Experimental results show that our proposed approach is effective in improving the performance of inductive link prediction, particularly in terms of ranking accuracy. The code and data used in our experiments are all available at https://github.com/Bohemianc/ISWC23-typed-inductive-LP.

2. Our Approach

In this section, we provide a detailed description of our solution that utilizes type information. First, we introduce the two forms of information we use, namely entity types and type hierarchies, and explain how we infer new entity types through type hierarchies. Second, we explain how we effectively integrate graph structures with type information, including details of joint training.

2.1. Type Mapping

We consider two forms of type information: (1) types of entities, such as (albert_einstein, rdf:type, Physicist) expressing that Einstein is a physicist, (2) and type hierarchies, such as (Physicist, rdfs:subClassOf, Scientist) expressing that physicists belong to the category of scientists. Let $\mathcal{E}$, $\mathcal{T}$, and $\mathcal{R}$ denote the entity set, type set, and relation set in the KG, respectively. Entity type assertions are defined as $(e, \text{rdf:type}, t)$ with $e \in \mathcal{E}$ and $t \in \mathcal{T}$, while type hierarchy assertions are defined as $(t_1, \text{rdfs:subClassOf}, t_2)$ with $t_1, t_2 \in \mathcal{T}$.

Additionally, we refer to (e_1, r, e_2) with $e_1, e_2 \in \mathcal{E}$ as an *instance triple*, and (t_1, r, t_2) with $t_1, t_2 \in \mathcal{T}$ as a *type triple*.

In contrast to most existing works utilising type information [3] [4], we use explicit entity type assertions, which are more accurate than learnable type representations. However, this leads to two issues: incomplete entity type information and type imbalance (i.e., the numbers of entities of different types can be diverse). We believe that the impact of type imbalance on type representation is caused by overly specific entity types. For example, if there are much more mathematicians than physicists in the KG, the model will consider the type triple (Mathematician, develops, Theory) to be more likely than (Physicist, develops, Theory), while (Scientist, develops, Theory) is a more reasonable type triple that is not affected by type imbalance. To address these two issues, we use type hierarchies to explicitly infer new entity type assertions. Specifically, we apply the inference rule

$$(e, \text{rdf:type}, s) \land (s, \text{rdfs:subClassOf}, t) \rightarrow (e, \text{rdf:type}, t).$$

We recursively use this rule to supplement entity type assertions to address the incomplete entity type issue. To address the type imbalance issue, we use type hierarchy assertions to select the most general type among an entity's multiple types, such as types Person and Location. Note that we do not consider the type Thing to prevent mapping all entities to this type.

2.2. Fusing Graph Structures with Type Information

Next, we introduce how to integrate graph structures with type information. For graph structures, we adopt the method proposed in RMPI [2], which is one of the state-of-art models in inductive KGC. This method transforms the sampled subgraph from the KG into another graph in which each node is an instance triple in the KG. Then, it applies a GNN to the transformed graph and obtain the final score using a linear layer. For type information, we assume that both relations and types during inference are already present during training, and thus we can represent relations and types using relation-specific and type-specific embeddings. Inspired by the translation-based principle in TransE [5], given a type triple (t_1, r, t_2), we expect that $\mathbf{t}_1 + \mathbf{r} \approx \mathbf{t}_2$, where $\mathbf{t}_1$, $\mathbf{t}_2$ and $\mathbf{r} \in \mathbb{R}^d$ are embeddings of types or relations. Although we map entities to general types, the types of an entity are not necessarily unique. For entities with multiple types, we take the average score of corresponding type triples to obtain the likelihood of the type triple being true. By fusing graph structures and type information, our method prioritize those instance triples that are more likely to adhere to the type constraints. The score of an instance triple (u, r, v) is calculated as

$$E_2(u, r, v) = \frac{1}{|\mathcal{T}(u)| \times |\mathcal{T}(v)|} \sum_{t_1 \in \mathcal{T}(u), t_2 \in \mathcal{T}(v)} -||\mathbf{t}_1 + \mathbf{r} - \mathbf{t}_2||_2, \tag{1}$$

$$\text{score}(u, r, v) = E_1(u, r, v) + E_2(u, r, v), \tag{2}$$

where E_1 is the score function of RMPI, E_2 is the score function of type triples, and $\mathcal{T}(e)$ is the set of types of an entity e.

Following previous works, we adopt margin ranking loss for training. Specifically, we apply margin ranking loss to the final score $\text{score}(u, r, v)$ and jointly train the two energy functions E_1 and E_2. An alternative training strategy is to separately train the energy functions,

Table 1
AUC-PR results on inductive link prediction.

Methods	FB15k237				NELL-995			
	v1	v2	v3	v4	v1	v2	v3	v4
GraIL	84.69	90.57	91.68	94.46	**86.05**	92.62	93.34	87.50
TACT	85.03	91.72	**93.14**	93.85	77.54	93.30	92.53	85.25
CoMPILE	**85.50**	91.68	93.12	**94.90**	80.16	**95.88**	**96.08**	85.48
RMPI-NE	85.22	92.08	91.77	92.27	81.07	93.64	94.99	88.82
Ours	84.22	**92.09**	91.67	92.77	77.79	94.23	95.67	**90.27**

Table 2
Hits@10 results on inductive link prediction.

Methods	FB15k237				NELL-995			
	v1	v2	v3	v4	v1	v2	v3	v4
GraIL	64.15	81.80	82.83	**89.29**	59.50	93.25	91.41	73.19
TACT	62.20	80.02	84.16	88.41	51.50	91.49	92.46	72.98
CoMPILE	67.66	82.98	84.67	87.44	58.38	93.87	92.77	75.19
RMPI-NE	**70.00**	82.85	83.18	86.52	**60.50**	94.01	91.78	**84.27**
Ours	68.78	**84.62**	**85.03**	89.22	**60.50**	**94.12**	**94.13**	84.06

similar to AutoETER [3]. However, optimizing E_2 alone using margin ranking loss can lead to issues, as the corresponding type triple of a negative instance triple may not necessarily be false. For example, the type triple $(\mathsf{Scientist}, \mathsf{develops}, \mathsf{Theory})$ of the negative triple $(\mathsf{feynman}, \mathsf{develops}, \mathsf{theory\_of\_relativity})$ still holds.

3. Experiments

We conducted experiments on two benchmark knowledge graphs, FB15k-237 and NELL995, each with 4 versions split by GraIL [1]. Since the existing datasets lack type information, we obtained original entity types from external knowledge graphs or entity names and completed them as detailed in Section 2.1. Specifically, for FB15k-237, we map the anonymous entities in FB15k-237 to entities in DBpedia by sameAs.org[1], and then retrieved entity types and type hierarchies through the meta-relations rdf:type and rdfs:subClassOf by querying DBpedia[2]. As for NELL995, its entity names inherently include a specific type of entities, and the associated website provides type hierarchies[3]. Our baselines include GraIL [1], TACT [6], CoMPILE [7] and RMPI-NE [2]. We select RMPI-NE, a variant of RMPI, as the representative of models proposed in [2] due to its superior performance than other variants. In our model, we also use RMPI-NE to calculate $E_1(u, r, v)$ in Equation 2. Our experiments aimed to demonstrate the effectiveness of our type-enhanced model in improving performance of inductive link prediction.

[1] http://sameas.org/store/freebase/
[2] https://dbpedia.org/sparql
[3] http://rtw.ml.cmu.edu/resources/results/08m/NELL.08m.1115.ontology.csv.gz

Table 1 and Table 2 present the evaluation results of inductive link prediction on AUC-PR (area under the precision-recall curve) and Hits@10 (the percentage of testing triples whose ground truths are ranked within top-10 positions), respectively. The results of four baselines are taken from [2]. The best results for each dataset are bold, and the second highest results are underlined. We can see that our type-enhanced model achieves competitive performance in terms of AUC-PR and outperforms the baselines on most datasets in terms of Hits@10. This suggests that the incorporation of type information effectively prioritizes instance triples that are more likely to adhere to the type constraints. The less impressive performance on the AUC-PR metric may be attributed to the presence of inaccuracies in the raw type information. Additionally, it is worth highlighting that our model consistently outperforms its base model RMPI-NE in most cases, both in terms of AUC-PR and Hits@10. This underscores the effectiveness of our approach in enhancing link prediction performance while emphasizing the need for further refinement in handling type information for more accurate results.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China under grant 61976153.

References

[1] K. Teru, E. Denis, W. Hamilton, Inductive relation prediction by subgraph reasoning, in: International Conference on Machine Learning, PMLR, 2020, pp. 9448–9457.

[2] Y. Geng, J. Chen, W. Zhang, J. Z. Pan, M. Chen, H. Chen, S. Jiang, Relational message passing for fully inductive knowledge graph completion, arXiv preprint arXiv:2210.03994 (2022).

[3] G. Niu, B. Li, Y. Zhang, S. Pu, J. Li, Autoeter: Automated entity type representation for knowledge graph embedding, arXiv preprint arXiv:2009.12030 (2020).

[4] J. Hao, M. Chen, W. Yu, Y. Sun, W. Wang, Universal representation learning of knowledge bases by jointly embedding instances and ontological concepts, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 1709–1719.

[5] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, Advances in neural information processing systems 26 (2013).

[6] J. Chen, H. He, F. Wu, J. Wang, Topology-aware correlations between relations for inductive link prediction in knowledge graphs, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, 2021, pp. 6271–6278.

[7] S. Mai, S. Zheng, Y. Yang, H. Hu, Communicative message passing for inductive relation reasoning, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, 2021, pp. 4294–4302.