Inductive Link Prediction in Knowledge Graphs: A Survey

Hongyu Sun*, Jian Luo*, Pengcheng Li*, Yue Shen*, Yulin Wang*

Abstract—Knowledge Graphs (KGs) are important resources containing structured knowledge in the form of triples and facilitate various applications. Due to the nature of the incompleteness of KGs, link prediction, the task of extending a KG with missing triples has attracted increasing attention. However, most of the existing work including KG embedding methods is limited to transductive link prediction where the same graph is used for training and inference. While inductive link prediction deals with missing triples involving components that are unseen during training. Inductive settings are more reasonable than transductive ones in practice because of the ever-evolving real-life KGs. Recent years have witnessed an emerging trend of inductive link prediction on KGs and a small amount of literature has been investigated to address this task. To our best knowledge, few investigations have looked into collecting and combing through relevant work. Thus we make the first try for a comprehensive survey, encompassing all the datasets, performance metrics, and methods. The methods are roughly divided into three categories, i.e., rule-based, GNN-based, and pretrain-based methods. Finally, we provide comparisons between different methods and possible promising direction after a meticulous analysis of the surveyed works.

Index Terms—inductive link prediction, knowledge graphs, rule, GNN, pre-trained language model

1 Introduction

Knowledge graphs (KGs) are aggregations of facts that designate the relationships between a group of entities that contain human knowledge. Because of the ability to model structured and complex data in a machine-readable way, KG is now widely used in various fields, such as question answering [40], semantic search [32], conversation generation [13], etc. These large KGs can encompass millions of entities and facts. However, most of them suffer from incompleteness, which limits their application in practice. Therefore, this paper focuses on the articles that address this challenge by performing a downstream task – inductive link prediction.

1.1 Background

In a knowledge graph, a fact is usually represented as a triplet format, i.e., (h, r, t) with h representing head entity, r representing relation and t representing tail entity. For example, in Figure 1, Da Vinci is the head entity of ”painted” and the Mona Lisa is the tail entity of the relation. Notable real-life examples of real-life KGs include FreeBase [3], WikiData [28], DBPedia [2], Yago [24]. Despite the marvelous power of KGs for modeling, most of the above knowledge graphs are incomplete, leading to the link prediction tasks on KGs. Link prediction, also called KG completion, aims to fill in the missing component in a triple, which can be subdivided into entity prediction and relation prediction. For example, if there exists an incomplete triple $(h, ?, t)$, link prediction, also called relation prediction in this case deals with filling in where the ”?” is.

A mainstream paradigm of link prediction is KG embedding which embeds entities and relations into low-dimensional spaces and measures the plausibility of a triple by defining a scoring function. However, KG embedding approaches and many other methods inherently assume that the set of entities in the graph is fixed and ignore the evolutionary nature of KGs. That is to say, real-world knowledge graphs are constantly evolving with new entities added, e.g., new users in social networking platforms, new products in online shopping platforms, and new molecules in medical graphs. These newly added entities can lead to the incompleteness of KG, i.e., missing new links. Much of the existing work focuses on transductive link prediction and cannot generalize to previously unseen entities [25]. In this case, already established models that fail to infer the new components need to be retrained, which is very costly. To address this challenge, inductive link prediction is proposed.

1.2 Task description and more details

A superiority of inductive link prediction is that the completion function learnt for a given KG is applicable to any other KG. In other words, a trained example of this task can predict triples involving constants not seen during training.

In inductive link prediction, different graphs can be used for inference, which includes either a combination of visible entities and invisible entities (i.e., semi-inductive) or only invisible entities (i.e., complete-inductive). While the subset of the connections are same with the training graph. In the evaluation process, there are two scoring tasks between entities during the training process [12]: seen-seen-induction and unseen-unseen-induction. The area of representation learning over link prediction in knowledge graphs has been dominated by one task: transductive link prediction. In the

- ** These five authors contributed equally.*
- *H. Sun, J. Luo, Y. Shen and Y. Wang are with the Department of Electronic Engineering and Information Science, University of Science and Technology of China (USTC), Hefei, China. (E-mail: sunhongyu@mail.ustc.edu.cn; jianluo@mail.ustc.edu.cn; yueshen@mail.ustc.edu.cn; yulinwang@mail.ustc.edu.cn).*
- *P. Li is with the Institute of Advanced Techonology, USTC, Hefei, China. (E-mail: pechola@mail.ustc.edu.cn).*

transductive setup, link prediction is performed to inference over the same graph seen at training time. The existing transduction models have three limitations : i) lock entity sets to be the same during at training and inference time. ii) allow shallow embedding of the model to learn the unique vector of each node in the graph. iii) cannot use a pre-trained model to reason about new graph. With the growth of graphs in industry and enormous computational costs of re-training every time a graph has changed, the focus of KG representation learning is moving towards inductive models that are free from the weakness in transductive models.

1.3 Structure of the survey

The objectives of this paper are two-fold. First, it is noticeable that there is no relevant literature to collect the growing knowledge about inductive link prediction in a knowledge graph. Our goal is to start with several basic concepts, provide a glimpse of the task of inductive link prediction in KG, and make a comprehensive comparison of publicly available datasets. Second, we explore almost all the state-of-the-art inductive link prediction methods in the knowledge graph and categorize them into three types: rule-based methods, GNN-based methods and pretrain-based methods. We present their key technologies and characteristics from varieties of perspectives.

The rest of this paper is organized as follows. Methodology in Section 2 is about the retrieval strategy and scope. Section 3 introduces inductive link prediction associated with notations, datasets, and performance metrics in the context of a knowledge graph. Section 4 reviews rule-based methods with their corresponding pros and cons. Comprehensive comparisons of the GNN-based methods in Section 5. Section 6 summarizes the pretrain-based methods with their characteristics. We conclude this paper and identify future research directions in Section 7. We also set up a regularly updated project page on: https://betterwyl.github.io.

2 Methodology

This review follows a rigorous descriptive systematic review approach, using the following steps to search and filter for relevant articles: i) identify articles relevant to this review in publication search engines such as Google Scholar and relevant conference proceedings, using "'inductive' AND ('knowledge graph' OR 'link prediction' OR 'reasoning')" as keywords to search articles related to this review. ii) Filtering by year of publication and name of the journal or conference in which the paper was published, the candidate papers were mainly published after 2020, but also included significant articles published before. The articles are mainly from ACL, AAAI, EMNLP, ICLR, ICML, IJCAI, KDD, NeurIPS, NAACL and WWW. iii) The articles are reviewed based on title, abstract, and keywords to ensure the validity and relevance of the study to the discussion of knowledge graph inference.

3 Definitions and Preliminaries

3.1 Knowledge graph

As mentioned in 1, a knowledge graph consists of many triples and each triple can be represented as (h, r, t). A knowledge graph can be defined as $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, where $\mathcal{E}$,$\mathcal{R}$ and $\mathcal{T} \subset \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ are the set of entities, relations and triplets.

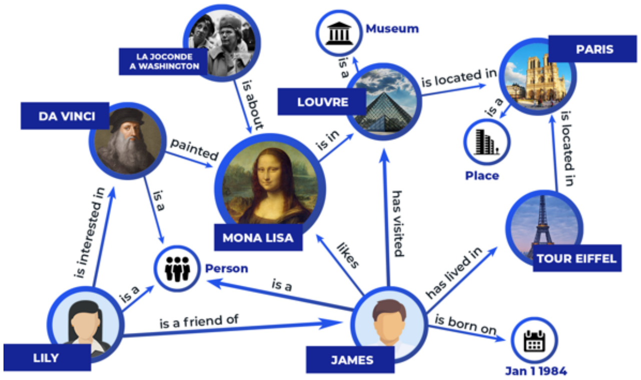


Fig. 1. Illustration of knowledge graphs.

3.2 link prediction

Link Prediction is the task of exploiting the existing facts in a knowledge graph to infer missing triples. This amounts to guessing the correct entity that completes $(?, r, t)$ (head prediction), $(h, r, ?)$ (tail prediction) and $(h, ?, t)$ (relation prediction).

3.3 Inductive link prediction

Let $\mathcal{G}_{tr} = (\mathcal{E}_{tr}, \mathcal{R}_{tr}, \mathcal{T}_{tr})$ denotes a training graph, where $\mathcal{E}_{tr}$, $\mathcal{R}_{tr}$ and $\mathcal{T}_{tr} \subset \mathcal{E}_{tr} \times \mathcal{R}_{tr} \times \mathcal{E}_{tr}$ are the set of entities, relations and triplets during training. The inductive link prediction task is defined as the training model on $\mathcal{G}_{tr}$, running inference over for any new graph $\mathcal{G}^{'} = (\mathcal{E}^{'}, \mathcal{R}^{'}, \mathcal{T}^{'})$. Relations in new model are all occurred in training, i.e., $\mathcal{R}^{'} \subseteq \mathcal{R}_{tr}$, the model can predict missing triplets in graph $\mathcal{G}^{'}$. i.e., $(?, r, t)$, $(h, r, ?)$ and $(?, r, t)$, where $h,\ t \in \mathcal{E}^{'}$ and $r \in \mathcal{R}^{'}$. The set of all entities in reality as $\mathcal{E}$ and the set of all knowledge graphs whose relation set $\mathcal{R}$ are a subset of $\mathcal{R}_{tr}$ and $\mathcal{T}_{tr}$. A scoring function used to train the model is :

$$f : \mathcal{G}_{tr} \times \mathcal{E} \times \mathcal{R}_{tr} \times \mathcal{E} \to \mathbb{R}$$
$$(\mathcal{G}^{'}, h, r, t) \mapsto f(\mathcal{G}^{'}, h, r, t)$$

where $\mathcal{G}^{'} = (\mathcal{E}^{'}, \mathcal{R}^{'}, \mathcal{T}^{'})$, $\mathcal{R}^{'} \subseteq \mathcal{R}_{tr}$ and $h,\ t \in \mathcal{E}^{'}$.

3.4 Datasets

The availability of large-scale datasets has been continuously fostering the community with the advent of data-driven era. As regards inductive link prediction, we summarize publicly available datasets in Table 1 , out of which the WN18RR [10] , FB15K-237 [26] , and NELL-995 [6] are the typical representatives. In a sequel, we selectively introduce these three datasets with regard to their size, diversity accordingly.

Dataset size. WN18RR is a subset of 40,943 entities and 11 relationships extracted from WordNet [20].FB15k-237 is a subset of the knowledge graph Freebase, there are 14,505 subject terms in the knowledge base and a total of 237 relationships. NELL-995 is a NELL subset that is suitable for multi-hop reasoning from the 995th iteration of the NELL system, and it has 75,492 entities and 200 relationships. The WN18RR, FB15k-237, and NELL-995 benchmark datasets were originally developed for the transductive setting.The entities of the standard test splits are a subset of the entities in the training splits. The statistics of these datasets are shown in .

TABLE 1
A summary of datasets for inductive link prediction in the context of knowledge graphs.

Datasets	Entities	Relationships
WN18RR	40,943	11
FB15k-237	14,505	237
NELL-995	75,492	200
UMLS	135	46
Wikidata-5M	4,579,609	822

Extension. In order to facilitate inductive testing, fully-inductive benchmark datasets by sampling disjoint subgraphs was created from the original datasets [25] .There are four versions of inductive datasets from each knowledge graph with increasing sizes.The statistics of these inductive benchmarks is given in Table 2 .

3.5 Performance metrics

Inductive link prediction in knowledge graph adopts the Area Under Curve(AUC) as its primary performance metrics, all of which stem from the same ideology in knowledge graph. We first reviewed the general form of AUC metrics, and then recognized the tricky connections and differences between dataset specific AUCs used in common benchmarks. The Hits at(H@k) metric denotes the relative frequency of the correct entity's rank being at most k, for different $k \in \{1, 3, 5, 10, 100\}$. Its value range is [0, 1].

4 Rule-based Methods

4.1 Introduction to Logical Rules

Some known facts in the knowledge graph can be implied with first-order logic rules, then these rules can be used to infer the facts that can't be obtained directly through the knowledge graph. For example, there is a rule

$$workIn(X, Y) \wedge locateIn(Y, Z) \rightarrow liveIn(X, Z) \tag{1}$$

meaning that if X works in Y and Y locates in Z, then the fact that X lives in Z can be inferred. Logical rules like this have strong interpretability, so we can apply these rules to domains and data which previously had not been seen. Chain-like logical rules are like

$$\alpha\ query(Z, X) \leftarrow R_n(Z, Y_n) \wedge \cdots \wedge R_1(Y_1, X) \tag{2}$$

$R_1 \ldots R_n$ denote relations, for short, we called $p = \wedge R_i$ a rule pattern. A rule path is composed of a sequence of entities according to a pattern p. The same pattern may correspond to different paths, like the path of $\{R_a(X, M), R_b(M, Y)\}$ differs from the path of $\{R_a(X, N), R_b(N, Y)\}$, but they are the same pattern. α is the rule's confidence, ranging from 0 to 1. When an entity x is given, we can calculate the sum of α that relates to each rule. All of these rules make contribute to $query(y, x)$, and each y is evaluated with the sum related to the query.

4.2 Rule Mining Methods

The objective of rule mining methods for knowledge graphs is first-order predicate logic rules on knowledge graphs, which usually consists of the form body => head. The rule body is composed of multiple atoms of parsing, and the rule head is a single atom containing the target predicate.

Rule mining on the knowledge graph can be used to describe the general laws of data. This helps to understand the data, and based on that, to reason, complement, and check and correct errors.

AMIE [11] is a classical rule mining model with the goal of mining concatenated, closed-form Horn rules, using three mining operations operators to iteratively extend the rules while defining multiple criteria to evaluate and prune the rules. AMIE+ [17] modifies AMIE by optimizing the query writing technique and improving the rule quality estimation. Although it is much more efficient than the former and can be used for large datasets, the learning efficiency is still insufficient.

The idea of the IterE [39] model is to introduce entity and attribute information in the rule mining process. The target of this mining approach is ontology axioms, and the authors selected seven expression axioms related to object attributes and transformed them into rule form. The iterative learning idea is used, where rules are learned from knowledge representations by a rational pruning strategy, and knowledge representations are learned from existing triples and rule-predicted triples.

Neural LP [35] is a classic method for learning logic rules with an end-to-end differentiable model. It is based on TensorLog [7], a system that makes reasoning differentiable and furthermore integrates knowledge into a learning system based on gradient. In the factor graph, differentiable functions are used to answer queries, which are composed of evidence variables and query variables. Some logical inference tasks can be compiled into sequences of differentiable numerical operations on matrices.

One defect of TensorLog is that it only takes parameters into consideration, rather than learning the rules. Fan Yang and Zhiling Yang come up with a neural logic programming framework [35], which overcomes the deficiency of TensorLog that only learns the parameters. Inductive logic programming always signifies the learning process involves proposing new logical rules. In this framework, an end-to-end differentiable model combines the structure and parameter learning of logical rules. Structure means the specific sets of rules included in a model, while parameter means the confidence associated with each rule.

Neural LP and Neural LP-based methods lack practical attempts in the situation of multiple objects connected by the same relation with one object. Multi-target Probabilistic Logic Reasoning (MPLR) [31] makes an improvement in the basis of Neural LP, allowing for more queries fed in one

TABLE 2
Statistics of inductive benchmark datasets.

		WN18RR			FB15k-237			NELL-995		
		#relations	#nodes	#links	#relations	#nodes	#links	#relations	#nodes	#links
v1	train	9	2746	6678	183	2000	5226	14	10915	5540
	ind-test	9	922	1991	146	1500	2404	14	225	1034
v2	train	10	6954	18968	203	3000	12085	88	2564	10109
	ind-test	10	2923	4863	176	2000	5092	79	4937	5521
v3	train	11	12078	32150	218	4000	22394	142	4647	20117
	ind-test	11	5084	7470	187	3000	9137	122	4921	9668
v4	train	9	3861	9842	222	5000	33916	77	2092	9289
	ind-test	9	7208	15157	204	3500	14554	61	3294	8520

batch and then reasoning in multi-target cases. This research firstly proposes two new indicators to evaluate KG reasoning tasks, saturation helps measure whether the learned rules are interpretable from another aspect, and bifurcation augments traditional metrics, computing the proportion of instances with multiple reasoning destinations. These two indicators are applied to several knowledge graph benchmarks to get a better understanding of their data structure and further help MPLR generates high-quality logic rules.

Dash and Goncalves design a linear programming (LP) formulation [9] and use the standard column generation method, which is used for classification, to solve the complex LP. The initial rules set take part in constructing the LP, which helps find the best subset of numerous rules with weights. After that, new rules are generated to supplement the current rule set. Restrictions used to limit the complexity of the rule set are added to ensure the interpretability of the new rules.

Another RLvLR [22] method performs rule mining by exploring the embedding space of predicates and parameters. RLvLR proposes a new sampling method to construct the loose parts, which greatly reduces the input. Meanwhile, it introduces parameter embedding to improve the rule quality estimation algorithm. It also proposes a scalable rule mining method with efficient rule search and pruning, and uses the extracted rules for link prediction.

Rule mining methods oriented to knowledge graphs have continued to receive widespread attention due to their advantages such as high rule interpretability and accuracy, but face problems such as poor scalability, complexity of rules and limited expressiveness.

4.3 Path representation methods

In order to further improve the relational reasoning based on embedded representation, some scholars have incorporated relational path information into it and achieved good results. However, there are some problems with the embedded representation approach of incorporating relational paths into knowledge bases. Although the use of relational paths can improve the performance of the model, it also poses a key technical challenge. Since the number of possible relational paths between entity pairs grows exponentially with the path length, the complexity of training increases dramatically. To address this problem, path-RNN [21] is approximated by sampling or pruning, selecting the paths that are closest to the type to be predicted. The path-RNN proposes a method to reason about the connections of multi-hop relations non-atomically, composing the meaning of a path with a recurrent neural network (RNN) that takes as input a vector embedding of binary relations in the path. But an RNN model is suitable for learning only one relation, which is not applicable in reality.

Another problem is that current methods only consider path information and not node information, even if the same path contains different nodes possessing different information. All-path [27] proposes a dynamic planning approach that optimizes for the time complexity of previous embedded representations incorporating path information and adds node information, aiming at efficient computation and incorporating more adequate information. All -PATH outperforms previous methods in terms of time and effectiveness. However, the implementation of ALL-PATH is only based on the bilinear model, which can be replaced by other models, applying this idea of relational path integration to many existing embedded representation learning methods.

5 GNN-BASED METHODS

5.1 Introduction to GNNs

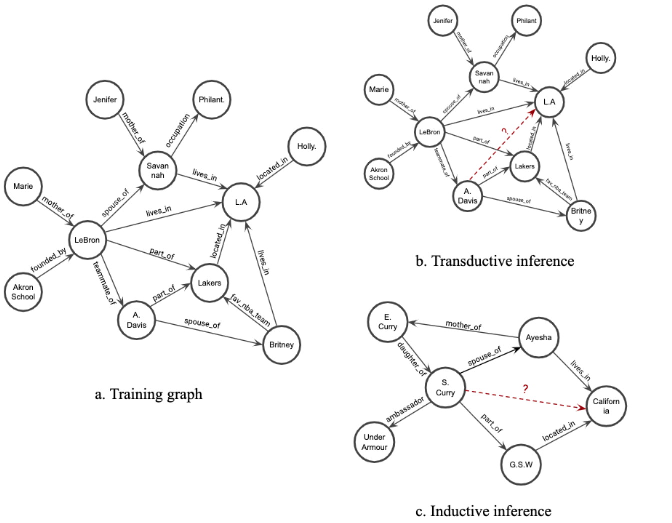
Graph Neural Network (GNN) is a complex neural model that encodes the topological structure of a graph into low dimensional vectors. GNN can capture the inherent topology information by mapping the adjacent nodes in the graph to the adjacent points in the embedded space. Aside from the KG literature, it is proved theoretically that GNN can learn common graph heuristic methods to predict links in simple graphs.

5.2 Subgraph-based Methods

Motivated by GNN aggregating local information, several inductive models based on GNN are proposed. Many works model the subgraph of the target triplet to capture the topological structure with inductive ability.

5.2.1 Subgraph-based Methods

GraIL [25] is the first method proposed to model enclosing subgraph structure around the target triple. It reasons over local subgraph structure and has a powerful inductive bias to learn entity-independent relational semantics. Unlike embedding-based models, GraIL is naturally inductive and can be extended to unseen entities and graphs after training.



a. Training graph

b. Transductive inference

c. Inductive inference

Fig. 2. Illustration of transductive and inductive settings for relation prediction in knowledge graphs.

The results show that GraIL is superior to the existing inductive rule baseline in inductive settings.

To handle the asymmetric and anti-symmetric triplets effectively, a novel Communicative Message Passing neural network for Inductive reLation rEasoning (CoMPILE) [18] was proposed. It reasons over locally directed subgraph structures and has a vigorous inductive bias to process entity-independent semantic relations. In CoMPILE, enclosing directed subgraph was firstly extracted for each triplet instead of an undirected one. Therefore, CoMPILE solves the problem of directionality of extracting subgraphs and the weak role of relational information in subgraph modeling in previous work.

Aiming at the problem that many existing inductive reasoning methods do not consider adjacent relational triples when predicting missing links, TACT [5] takes advantage of the neighboring relational triples. It can effectively exploit topology-aware correlaTions between relations in an entity-independent manner. Specifically, relation pairs are categorized into several topological patterns, and a Relational Correlation Network is proposed to learn the importance of the different patterns for inductive link prediction. This work proves that the semantic correlation between the two relationships is highly related to their topological structure in the knowledge graph.

SNRI [33]consider that the enclosing part of the subgraph without complete neighboring relations causes two challenging problems. Not only enclosing subgraph may be empty or sparse but also partial neighboring relations will lose due to the nature of the enclosing subgraph. To work well in that case, SNRI comprehensively establishes complete neighboring relations from two aspects: neighboring relational feature for node feature and neighboring relational path for a sparse subgraph. Mutual information (MI) maximization is the core technique in the paper for the knowledge graph to further model neighboring relations globally.

5.2.2 Modeling graph structures

To model the graph structure around the triple (u, r_t, v), this task can be divided into three parts : (i) extracting the enclosing subgraph around the target nodes, (ii) labeling the nodes or the edge in the extracted subgraph, and (iii) scoring the labeled subgraph using a GNN.

Step 1: subgraph extraction. The local graph neighborhood of a particular triplet assumed in the KG will contain the logical evidence needed to deduce the relationship between the target nodes. In particular, paths connecting the two target nodes contain the information that could imply the target relation. The enclosing subgraph around the target nodes. Define the *enclosing subgraph* between nodes u and v as the graph induced by all the nodes that occur on a path between u and v. It is given by the intersection of neighbors of the two target nodes followed by a pruning procedure. More precisely, let $\mathcal{N}_k(u)$ and $\mathcal{N}_k(v)$ be a set of nodes in the k-hop (undirected) neighborhood of the two target nodes in the KG. Compute the enclosing subgraph by taking the intersection, $\mathcal{N}_k(u) \cap \mathcal{N}_k(v)$, of these k-hop neighborhood sets and then prune nodes that are isolated or at a distance greater than k from either of the target nodes. To handle the direction of the target triplet, CoMPILE uses a directed enclosing subgraph. SNRI reserve complete neighboring relations $\mathcal{N}^r(u)$ of each node, which contains relations partially omitted by enclosing subgraph.

Step 2: Node/Edge Labeling Initialization. Initialize the node embedding by the distances to the target head and target tail to capture the relative position of each node in the subgraph. A node feature matrix, $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times d_i}$, as input, which is used to execute the neural message passing algorithm in GraIL. Each node, i, in the subgraph around nodes u and v is labeled with the tuple $(d(i,u), d(i,v))$, where $d(i,u)$ denotes the shortest distance between nodes i and u without counting any path through v (likewise for $d(i,v)$). This captures the topological position of each node with respect to the target nodes and reflects its structural role in the subgraph. The two target nodes, u and v, are uniquely labeled $(0,1)$ and $(1,0)$ so as to be identifiable by the model. The node features are thus

$$N_i = [\text{one-hot}(d(i,u)) \oplus \text{one-hot}(d(i,v))], \tag{3}$$

where $\oplus$ denotes concatenation of two vectors. Note that as a consequence of Observation 1, the dimension of node features constructed this way is bounded by the number of hops considered while extracting the enclosing subgraph.

In For edge I, i.e. (u_i, r_i, v_i), the initialized edge embedding is defined as $E_i = N_{u_i} \oplus E_{r_i} \oplus N_{v_i} \in \mathbb{R}^{4(u+2)\times d}$.

Step 3: GNN scoring . Scoring function in GNN is used to measure the ability of the subgraph inductive learning. An asymzmetric scoring function are designed in[25] by concatenating four related vectors. The subgraph representation $\mathbf{h}^L_{\mathcal{G}_{(u,v,r_t)}}$, the target nodes' latent representations $\mathbf{h}^L_u$ and $\mathbf{h}^L_v$, and a learned embedding of the target relation $\mathbf{e}_{r_t}$—and pass these concatenated representations through a linear layer:

$$\text{score}(u, r_t, v) = \mathbf{W}^T[\mathbf{h}^L_{\mathcal{G}_{(u,v,r_t)}} \oplus \mathbf{h}^L_u \oplus \mathbf{h}^L_v \oplus \mathbf{e}_{r_t}]. \tag{4}$$

To deal with asymmetric and anti-symmetric relations, CoMPILE preserve directed nature as well as to be consistent with the definition of the edge information. f denotes nonlinear activation function to increase the nonlinear modeling

TABLE 3
AUC-PR results on the inductive benchmark datasets extracted from WN18RR, FB15k-237 and NELL-995.

	WN18RR				FB15k-237				NELL-995			
	v1	v2	v3	v4	v1	v2	v3	v4	v1	v2	v3	v4
Neural LP	86.02	83.78	62.90	82.06	69.64	76.55	73.95	75.74	64.66	83.61	87.58	85.69
DRUM	86.02	84.05	63.20	82.06	69.71	76.44	74.03	76.20	59.86	83.99	87.71	85.94
GraIL	94.32	94.18	85.80	92.72	84.69	90.57	91.68	94.46	86.05	92.62	93.34	87.50
CoMPILE	98.23	99.56	93.60	99.80	85.50	91.68	93.12	94.90	80.16	95.88	96.08	85.48
TACT	96.15	97.95	90.58	96.15	88.73	94.20	97.10	98.30	94.87	96.58	95.70	96.12
SNRI	99.10	99.92	94.90	99.61	86.69	91.77	91.22	93.37	-	-	-	-

capacity of the model.The scoring function is changed as below:

$$\text{score}(u, r_t, v) = f[\mathbf{N}_u^L + \mathbf{R}_{r_t} - \mathbf{N}_v^L] \tag{5}$$

TACT organizes relational correlation module output $\mathbf{r}_t^F$ and graph structure module output $\mathbf{e}_S$in a unified framework, so the scoring function is adapted as follows:

$$\text{score}(u, r_t, v) = \mathbf{W}^T[\mathbf{r}_t^F \oplus \mathbf{e}_S] \tag{6}$$

In SNRI, the enclosing subgraph information and neighboring relational path information $\mathbf{p}_G$ as the final representation of subgraph score

$$\text{score}(u, r_t, v) = [\mathbf{h}^L_{\mathcal{G}_{(u,v,r_t)}} \oplus \mathbf{p}_G] \tag{7}$$

5.3 Other Methods

Despite that subgraph-based methods can be applied in inductive settings, they suffer from the high computational cost of generating a subgraph and re-training for every relation to be predicted, which is impractical in real scenes and is not scalable to large graphs. The new fully-inductive benchmark datasets created by GraIL [25] are rather small where the largest one FB15k-237/v4 has 34k training triples and 14k inferencing triples. Therefore, there are some other methods based on GNNs which are out of the current stream of subgraph-based methods and we will introduce them as follows.

Instead of learning node representation and graph representation as previous models, some recent works encode the original graph into an advanced graph and build a GNN framework on the advanced graph. For example, INDIGO [15] transforms the input KG into a node-annotated graph where the process of encoding is complicated and can be depicted in the figure[]. Then the authors utilize a GNN variant, i.e., GCN to aggregate local information of each node in this advanced graph and update the annotations of the graph nodes. The predicted triples can be read out from the outer layer directly instead of defining a scoring function additionally, which facilitates processing the entire graph at once and avoid the drawbacks of GraIL – re-training during prediction of a new triple. Motivated by rule-based methods and knowledge in mathematics of algebraic topology that the space of cycles is a vector space under certain assumptions, (cycleGNN, [34]) regard logical rules in KGs as cycles and learn rules in the space of cycles. They propose a novel framework, named cycleGNN which applys GNNs to the collected cycles and learns the representation of cycles. Experiments demonstrate the effectiveness of using cycles to represent rules and cycleGNN outperforms other benchmarks in relation prediction.

Before cycleGNN, (NBFNet, [42]) leverage the generalized sum of embeddings of all paths between a pair of nodes as representations of the two nodes. Inspired by the classic Bellman-Ford algorithm for solving the shortest path problem, they generalize this algorithm and build a model, named Neural Bellman-Ford Network(NBFNet) to solve the path formulation in the generalized Bellman-Ford algorithm. Despite that NBFNet focuses on representing paths between a pair of nodes, we classified this work into GNN-based methods because it is inherently a GNN framework which can be abstracted into a source-specific message passing process. Algorithm 1 outlines the pseudo-code of NBFNet. In addition to its generalization in the inductive settings, this work is worthy of attention for its interpretability, scalability, and excellent performance compared to previous works.

6 Pretrain-based Methods

6.1 Introduction to Pre-trained Language Model

Pre-trained language model (PLM) leverages the large corpus and achieves great success in many natural language processing down stream tasks. Large pre-trained laguage models like GPT3[4], BERT[1], XLNet[36], T5[23], RoBERTa[16] and ALBERT[14], are first trained on large unlabeled corpus on the pre-train stage, and then adapt to the down stream tasks by fine-tuning the model. PLMs capture the knowledge in large corpus with the huge parameters and sophisticated model structure, which benefit PLMs' generalization. Rich knowledge captured from large unlabeled corpus allows the model to achieve amazing results on few-shot even zero-shot scenario.

Textual information in the knowledge graph with appropriate training methods transfers the inference task on graphs to the sentence prediction task. While it is also available to contrastive learning approach, which enable the use of large pre-trained models. It makes the pretrain-based methods inherently gain the inductive reasoning power.

6.2 Pre-trained Language Model for Inductive Link Prediction in Knowlegde Graph

Knowledge graphs consist of millions of triplets, which are semantically enhanced with a textual description of entities

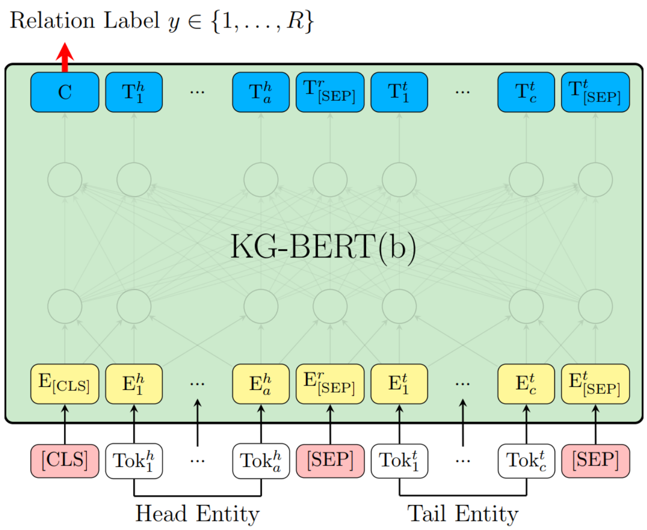


Fig. 3. Illustration of KG-BERT from [37]

and relations. Despite the success of the existing rule-based or GNN-based approach, which leverages the structure of the knowledge graph, textual information of the knowledge graph is ignored in such methods.

The great success of pre-trained language models in many natural language processing tasks has led us to think about how to apply it to knowledge graph tasks. Textual description of entities and relations. Text descriptions of entities and relations give us an opportunity to take advantage of pre-trained language models.

Pretrain-based approach leverages text descriptions of entities and relations and infers from the textual information. Pretrain-based methods have attracted attention in recent years and achieved better performance compared to many GNN-based and Rule-based methods.

KG-BERT[37] is the first work that introduces the large pre-trained language model into the link prediction task. KG-BERT treats the link prediction task as a multi-classification task, while the output is the probability of each entity. Same as the previous works, KG-BERT also adapts the ranking metrics to evaluate the result due to the open-world assumption. But different from rule-based or GNN-based methods, KG-BERT leverages the textual description of entities. The input of the KG-BERT is the concatenation of the triplet's textual description. Unseen entities and relations have their corresponding embedding consisting of token embeddings. Thus KG-BERT is available in the inductive scenario.

Compared to the KGC (Knowledge Graph Completion) model based on graph data, SimKGC [30] leverages the textual information about the entities and relations. SimKGC adapts contrastive learning to address the link prediction challenge, which allows the model to output the embedding representation of the entity similar to the traditional KGC methods. However, SimKGC takes the concatenation of head and relation as the input to the one BERT, while taking the tail as input to another BERT. The output of the SimKGC is the cosine-similarity score between two output vectors indicating the score of the triplet, which makes the SimKGC different from the existing KGC methods.

Rule-based method is explainable, which leverages the probability of each path in the graph and enables the explainable power of the model. Similar to the rule-based methods, BERTRL [38] utilizes the textual information and the graph structure. BERTRL combines the concatenation of the textual description of the triplet and an additional sentence of the path described in the knowledge graph. As mentioned in section 5.2.2, the subgraph extracted from the knowledge graph contains the target triplet (h, r, t). Path limited to length k from the head to the tail in the subgraph also extracted by the BERTRL. The path extracted from the subgraph is also called the reasoning path. Each reasoning path from h to t can be formulates in the form of $(h, r_0, e_1) \wedge (e_1, r_1, e_2) \wedge \cdots \wedge (e_n, r_n, t)$. The input is the concatenation of the description of the triplet and the textual sequence of the reasoning path. Owing to the textual sequence of reasoning path, BERTRL is able to conduct explainable reasoning by scoring the triple with different reasoning path.

Similar to SimKGC, StAR [29] takes the concatenation of textual description of heads and relations as the input to one pre-trained language model while takes the textual description of the tail to another PLM. Same as the SimKGC, these two PLM encoders do not share the parameters. But StAR hybrids the PLM and graph embedding approach, which benefits from the textual information and structure information. In the PLM forward stage, the output embedding of the concatenation u and the output embedding of the tail v are both contextualized representations. And two scoring strategies parallel in the second forward stage. One is designed for classification and another is designed to simulate the geometric computation in the low-dimension embedding space. The idea of the latter strategy follows the traditional knowledge graph completion model. Based on StAR, StATIK [19] proposes a structure composed of a GNN and the BERT. GNN leverages the structure information and the BERT leverages the textual information. Thus StATIK utilizes the structure and the textual information in one framework.

7 Conclusion

7.1 Concluding Remarks

This paper is the first to thoroughly survey the existing approaches on inductive link prediction in the context of KG, and consequently facilitates the audiences either to continue the research or to get inspired from it.

Depending on the characteristics of each approach, we roughly structure existing methods along three dimensions: (1) rule-based, (2) GNN-based, and (3) pretrain-based, allowing us to systematically review the existing work on the inductive link prediction and offering a clear perspective of which problems have been solved and what remains to be done. 18 well-known inductive link prediction models are compared with those classified in the same type in our survey.

7.1.1 Mining the Rules

In terms of the explainable power of rules, which enables the model to structure the first-order logic query explicitly, rule-based methods take the first lead to propose the solution in a heuristic way. AMIE [11], an exceeding pioneer, proposes

TABLE 4
Comparison of Inductive Link Prediction Methods

Type	Method	Inductive Settings		Explainable	Structure Information
		Unseen Entity	Unseen Relation		
Rule-based	AMIE[11]	✓	×	×	×
	Neural LP[35]	✓	×	✓	×
	IterE[39]	✓	×	✓	×
	path-RNN[21]	✓	×	✓	×
	All-path[27]	✓	×	✓	×
	RLvLR[22]	✓	×	✓	×
GNN-based	GraIL[25]	✓	×	×	✓
	ComPILE[18]	✓	×	×	✓
	TACT[5]	✓	×	×	✓
	SNRI[33]	✓	×	×	✓
	INDIGO[15]	✓	×	×	✓
	cycleGNN[34]	✓	×	×	✓
	NBFNet[42]	✓	×	✓	✓
Pretrained-based	KG-BERT[37]	✓	✓	×	×
	SimKGC[30]	✓	✓	×	✓
	BERTRL[38]	✓	✓	✓	×
	StAR[29]	✓	✓	✓	✓
	StATIK[19]	✓	✓	✓	✓

three operators as the basic operators to mine the rules behind the KG. Neural LP [35] leverages the end-to-end differentiable approach based on TensorLog [8], which integrates the knowledge by making the reasoning step differentiable. Despite the success of NeurLP, NeuralLP-like pipelines are short of a 1-to-N link prediction challenge. MPLR [31] makes further steps beyond, which broadens the advantages of rule-based methods to 1-to-N cases.

7.1.2 GNN for Inference

Structure information captured from heterogeneous graphs advances the inference power of GNN. GraIL [25] as the first of this line of work, inspired by NeurLP, proposes a subgraph-based approach. GraIL extracts the subgraph for the target triple. Each vertex is labeled by the distance between the head node and the tail node. Being free from entity embeddings injects inductive bias into the GraIL. To address the asymmetric/anti-asymmetric problem, CoMPILE [18] extracts the directed subgraph from head to tail. Only directed edges on the path from the head to tail follow the direction. Different from GraIL, TACT [5] proposes a line-graph enhanced GNN framework, which shows superior power on inductive link prediction. TACT leverages the connection between relations, which advances the inference on the knowledge graph. NBFNet [42] opens a new path of GNN-based approach. NBFNet formulates the general path framework inspired by the Bellman-Ford algorithm and applies the idea to the GNN. Experiments show that NBFNet exceeds the existing rule-based and GNN-based methods.

7.1.3 PLM as the Backbone

Text descriptions of entities and relations provide additional information, which is difficult for rule-based methods and GNN-based methods to exploit. Pre-trained language models can process text information well and take advantage of it. Thus, it is natural to think of leveraging pre-trained language models to address inductive link prediction challenges. KG-BERT [37] takes the first attempt to tackle the problem. KG-BERT transforms the link prediction task into the sentence prediction task. While SimKGC [30] introduces contrastive learning into the inductive link prediction task. Each of these two approaches has its own unique insights on the semantic information. StAR [29] takes a different approach, choosing to combine explicit graph structure information and semantic information as the input. This approach effectively takes advantage of both modals. However, how to introduce interpretability into pretrain-based methods is a big challenge. To address this challenge, BERTRL propose a path

7.2 Future direction

In the above parts of this survey, we have discussed the characteristics and the pros and cons of each specific inductive link prediction method. The in-depth conclusion and comparative analysis have provided evidence for the direction of future research work to some extent. In this subsection, we give our observations, thoughts, and predictions about the research trends in this area.

7.2.1 Better explainability

Despite the inferior performance of most of the rule-based methods, they can explicitly show the first-logic query and thus have an inherent explainability. Subgraph-based methods using GNNs predict missing relations by extracting a special subgraph, which can be regarded as the set of all paths connecting the head and tail entity. While during the process of inference, all of the subgraph information are exploited instead of one specific path and we cannot observe which path contributes most directly. Only NBFNet [42] in other methods can give a reasonable path and is explainable. Large pretrain language models lack interpretability as well. However, explainability of the model is necessary for the neural reasoning field which guarantees the availability of the proposed method. Therefore, one of the future goals of inductive link prediction is to make the inference in this setting more interpretable.

7.2.2 Better scalability

In fact, the first datasets built for inductive link prediction come from [25], whose size is quite small as can be seen from Table 2. The mentioned rule-based methods and GNN-based methods are both difficult to scale up to large graphs. NBFNet, a novel GNN framework, further improves scalability by single-source message passing and achieves state-of-start results. However, the model still fails to manage large graphs like Wiki-data [28]. Only with great scalability, can the proposed methods apply well in real-life scenarios.

7.2.3 Combination of structure and semantic information

We have shown that graph structure plays an important role in inductive link prediction on KGs, thus facilitating the performance of prediction using GNN-based methods compared to rule-based ones Table 3. In virtue of pre-trained language models, pretrain-based models incorporate external textual information to aid in prediction from the semantic dimension. As we all know, a knowledge graph is a graph containing human knowledge, where both structure and semantic information have large influences on the construction of itself. However, most of the existing work (either GNN-based methods or pretrain-based methods) is limited to a single type of information, neglecting the combination of them. In addition, it is worth mentioning that there exists a work combining BERT and GNN to accomplish tasks of node classification in homogeneous graphs [41]. Though the task is distinctively different from inductive link prediction in heterogeneous KGs, we believe that leveraging all of the available information is a promising trend. What's more, in SimKGC [30], the authors claim that the model leverages the structure information by proposing a graph-based re-ranking strategy that considers k-hop neighbors. It is a good try and there should be more methods that focus on incorporating different information in a more principled and natural way.

Acknowledgments

We would like to give our heartfelt thanks to all the people who have ever helped us.

Firstly, we would like to thank our English teacher, Mr. Xiong, for providing methodological skills for essay writing employed in this survey. We are also impressed by his conscientiousness. He held many discussions with us during the course, from which we gained a lot.

We also express our thanks to responsible and helpful assistant teachers for their hard work in this course.

Finally, the survey cannot be completed without our teammates' cooperation and persistence.

References

[1] Shivaji Alaparthi and Manit Mishra. "Bidirectional Encoder Representations from Transformers (BERT): A sentiment analysis odyssey". In: *CoRR* abs/2007.01127 (2020). arXiv: 2007.01127. URL: https://arxiv.org/abs/2007.01127.

[2] Sören Auer et al. "Dbpedia: A nucleus for a web of open data". In: *The semantic web*. Springer, 2007, pp. 722–735.

[3] Kurt Bollacker et al. "Freebase: a collaboratively created graph database for structuring human knowledge". In: *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. 2008, pp. 1247–1250.

[4] Tom B. Brown et al. "Language Models are Few-Shot Learners". In: (2020). arXiv: 2005.14165 [cs.CL].

[5] Jiajun Chen et al. "Topology-Aware Correlations Between Relations for Inductive Link Prediction in Knowledge Graphs". In: *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, 2021, pp. 6271–6278.

[6] Wenhu Chen et al. "Variational Knowledge Graph Reasoning". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*. Ed. by Marilyn A. Walker, Heng Ji, and Amanda Stent. Association for Computational Linguistics, 2018, pp. 1823–1832.

[7] William W. Cohen. "TensorLog: A Differentiable Deductive Database". In: *CoRR* abs/1605.06523 (2016).

[8] William W. Cohen, Fan Yang, and Kathryn Mazaitis. "TensorLog: A Probabilistic Database Implemented Using Deep-Learning Infrastructure". In: *J. Artif. Intell. Res.* 67 (2020), pp. 285–325. DOI: 10.1613/jair.1.11944. URL: https://doi.org/10.1613/jair.1.11944.

[9] Sanjeeb Dash and Joao Goncalves. "LPRules: Rule Induction in Knowledge Graphs Using Linear Programming". In: *CoRR* abs/2110.08245 (2021).

[10] Tim Dettmers et al. "Convolutional 2D Knowledge Graph Embeddings". In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*. Ed. by Sheila A. McIlraith and Kilian Q. Weinberger. AAAI Press, 2018, pp. 1811–1818.

[11] Luis Antonio Galárraga et al. "AMIE: association rule mining under incomplete evidence in ontological knowledge bases". In: *Proceedings of the 22nd international conference on World Wide Web*. 2013, pp. 413–422.

[12] Mikhail Galkin, Max Berrendorf, and Charles Tapley Hoyt. "An Open Challenge for Inductive Link Prediction on Knowledge Graphs". In: *CoRR* abs/2203.01520 (2022).

[13] He He et al. “Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings”. In: *arXiv preprint arXiv:1704.07130* (2017).

[14] Zhenzhong Lan et al. “ALBERT: A Lite BERT for Self-supervised Learning of Language Representations”. In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL: https://openreview.net/forum?id=H1eA7AEtvS.

[15] Shuwen Liu et al. “INDIGO: GNN-Based Inductive Knowledge Graph Completion Using Pair-Wise Encoding”. In: *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*. Ed. by Marc’Aurelio Ranzato et al. 2021, pp. 2034–2045.

[16] Yinhan Liu et al. “RoBERTa: A Robustly Optimized BERT Pretraining Approach”. In: *CoRR* abs/1907.11692 (2019). arXiv: 1907.11692. URL: http://arxiv.org/abs/1907.11692.

[17] Luis et al. “Fast rule mining in ontological knowledge bases with AMIE +”. In: *Vldb Journal* (2015).

[18] Sijie Mai et al. “Communicative Message Passing for Inductive Relation Reasoning”. In: *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, 2021, pp. 4294–4302.

[19] Elan Markowitz et al. “StATIK: Structure and Text for Inductive Knowledge Graph Completion”. In: *Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022*. Ed. by Marine Carpuat, Marie-Catherine de Marneffe, and Iván Vladimir Meza Ruíz. Association for Computational Linguistics, 2022, pp. 604–615. DOI: 10.18653/v1/2022.findings-naacl.46. URL: https://doi.org/10.18653/v1/2022.findings-naacl.46.

[20] George A. Miller. “WordNet: A Lexical Database for English”. In: *Commun. ACM* 38.11 (1995), pp. 39–41.

[21] Arvind Neelakantan, Benjamin Roth, and Andrew McCallum. “Compositional vector space models for knowledge base completion”. In: *arXiv preprint arXiv:1504.06662* (2015).

[22] Pouya Ghiasnezhad Omran, Kewen Wang, and Zhe Wang. “Scalable Rule Learning via Learning Representation.” In: *IJCAI*. 2018, pp. 2149–2155.

[23] Colin Raffel et al. “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer”. In: *J. Mach. Learn. Res.* 21 (2020), 140:1–140:67. URL: http://jmlr.org/papers/v21/20-074.html.

[24] 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*. 2007, pp. 697–706.

[25] Komal K. Teru, Etienne G. Denis, and William L. Hamilton. “Inductive Relation Prediction by Subgraph Reasoning”. In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 9448–9457.

[26] Kristina Toutanova and Danqi Chen. “Observed versus latent features for knowledge base and text inference”. In: *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality, CVSC 2015, Beijing, China, July 26-31, 2015*. Ed. by Alexandre Allauzen et al. Association for Computational Linguistics, 2015, pp. 57–66.

[27] Kristina Toutanova et al. “Compositional learning of embeddings for relation paths in knowledge base and text”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2016, pp. 1434–1444.

[28] Denny Vrandečić and Markus Krötzsch. “Wikidata: a free collaborative knowledgebase”. In: *Communications of the ACM* 57.10 (2014), pp. 78–85.

[29] Bo Wang et al. “Structure-Augmented Text Representation Learning for Efficient Knowledge Graph Completion”. In: *WWW ’21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*. Ed. by Jure Leskovec et al. ACM / IW3C2, 2021, pp. 1737–1748. DOI: 10.1145/3442381.3450043. URL: https://doi.org/10.1145/3442381.3450043.

[30] Liang Wang et al. “SimKGC: Simple Contrastive Knowledge Graph Completion with Pre-trained Language Models”. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Association for Computational Linguistics, 2022, pp. 4281–4294. DOI: 10.18653/v1/2022.acl-long.295. URL: https://doi.org/10.18653/v1/2022.acl-long.295.

[31] Yuliang Wei et al. “MPLR: a novel model for multi-target learning of logical rules for knowledge graph reasoning”. In: *CoRR* abs/2112.06189 (2021).

[32] Chenyan Xiong, Russell Power, and Jamie Callan. “Explicit semantic ranking for academic search via knowledge graph embedding”. In: *Proceedings of the 26th international conference on world wide web*. 2017, pp. 1271–1279.

[33] Xiaohan Xu et al. “Subgraph Neighboring Relations Infomax for Inductive Link Prediction on Knowledge Graphs”. In: *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*. Ed. by Luc De Raedt. ijcai.org, 2022, pp. 2341–2347.

[34] Zuoyu Yan et al. “Cycle Representation Learning for Inductive Relation Prediction”. In: *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*. Ed. by Kamalika Chaudhuri et al. Vol. 162. Proceedings of Machine Learning Research. PMLR, 2022, pp. 24895–24910.

[35] Fan Yang, Zhilin Yang, and William W. Cohen. “Differentiable Learning of Logical Rules for Knowledge Base Reasoning”. In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon et al. 2017, pp. 2319–2328.

[36] Zhilin Yang et al. "XLNet: Generalized Autoregressive Pretraining for Language Understanding". In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. Ed. by Hanna M. Wallach et al. 2019, pp. 5754–5764. URL: https://proceedings.neurips.cc/paper/2019/hash/dc6a7e655d7e5840e66733e9ee67cc69-Abstract.html.

[37] Liang Yao, Chengsheng Mao, and Yuan Luo. "KG-BERT: BERT for Knowledge Graph Completion". In: *CoRR* abs/1909.03193 (2019). arXiv: 1909.03193. URL: http://arxiv.org/abs/1909.03193.

[38] Hanwen Zha, Zhiyu Chen, and Xifeng Yan. "Inductive Relation Prediction by BERT". In: *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*. AAAI Press, 2022, pp. 5923–5931. URL: https://ojs.aaai.org/index.php/AAAI/article/view/20537.

[39] Wen Zhang et al. "Iteratively learning embeddings and rules for knowledge graph reasoning". In: *The World Wide Web Conference*. 2019, pp. 2366–2377.

[40] Yuyu Zhang et al. "Variational reasoning for question answering with knowledge graph". In: *Thirty-second AAAI conference on artificial intelligence*. 2018.

[41] Jianan Zhao et al. "Learning on Large-scale Text-attributed Graphs via Variational Inference". In: *arXiv preprint arXiv:2210.14709* (2022).

[42] Zhaocheng Zhu et al. "Neural Bellman-Ford Networks: A General Graph Neural Network Framework for Link Prediction". In: *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*. Ed. by Marc'Aurelio Ranzato et al. 2021, pp. 29476–29490.