A RETRIEVE-AND-READ FRAMEWORK FOR KNOWLEDGE GRAPH REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Knowledge graph (KG) reasoning aims to infer new facts based on existing facts in the KG. Recent studies have shown that using the graph neighborhood of a node via graph neural networks (GNNs) provides more useful information compared to just using the query information. Conventional GNNs for KG reasoning follow the standard message-passing paradigm on the entire KG, which leads to over-smoothing of representations and also limits their scalability. At a large scale, it becomes computationally expensive to aggregate useful information from the entire KG for inference. To address limitations of existing KG reasoning frameworks, we propose a novel retrieve-and-read framework, which first retrieves a relevant subgraph context for the query and then jointly reasons over the context and the query with a high-capacity reader. As part of our exemplar instantiation for the new framework, we propose a novel Transformer-based GNN as the reader, which incorporates graph-based attention structure and cross-attention from deep fusing between query and context. This design enables the model to focus on salient subgraph information that is relevant to the query. Empirical experiments on two standard KG reasoning datasets demonstrate the competitive performance of the proposed method.[1]

1 INTRODUCTION

Knowledge graphs encode a wealth of structured information in the form of "(subject, relation, object)" triples. The rapid growth of KGs in recent years has led to their wide use in diverse applications such as information retrieval (Castells et al., 2007; Shen et al., 2015) and data mining (Zheng et al., 2021). KG reasoning, which is commonly modeled as link prediction (Bordes et al., 2013) that aims to infer new facts based on existing facts, is a fundamental task on KGs. It finds applications in relation extraction (Wang et al., 2014; Weston et al., 2013), question answering (Bordes et al., 2014) and recommender systems (Zhang et al., 2016).

Early methods for KG link prediction have focused on learning a dense embedding for each entity and relation in the KG, which are then used to calculate the plausibility of new facts via a simple scoring function (e.g., cosine similarity) (Bordes et al., 2013; Lin et al., 2015; Ji et al., 2015; Socher et al., 2013; Dettmers et al., 2018). The hope is that an entity's embedding will learn to compactly encode the structural and semantic information in its neighborhood in a way that a simple scoring function would suffice for making accurate link predictions. However, it is challenging to fully encode the rich information of KGs into such shallow embeddings. Similar to how contextualized encoding models like BERT (Devlin et al., 2018) have been replacing static embeddings (Mikolov et al., 2013; Pennington et al., 2014) for natural language representation, several recent studies have adapted message-passing graph neural networks (GNNs) for KG reasoning (Schlichtkrull et al., 2018; Shang et al., 2019; Vashishth et al., 2019). By using a higher-capacity GNN to iteratively encode increasingly larger graph neighborhood, GNN-based KG reasoning methods have shown a great success. However, the reliance on message passing over the entire KG limits their scalability to large-scale KGs such as Wikidata (Vrandečić & Krötzsch, 2014). The same reason also leads to slow inference speed.

Intuitively, for each specific query, e.g., *(Barack Obama, collaborate_with, ?)*, only a small subgraph of the entire KG may be relevant for answering the query (Figure 1). If we could *retrieve* the

[1] All the code and data will be released on GitHub upon acceptance.

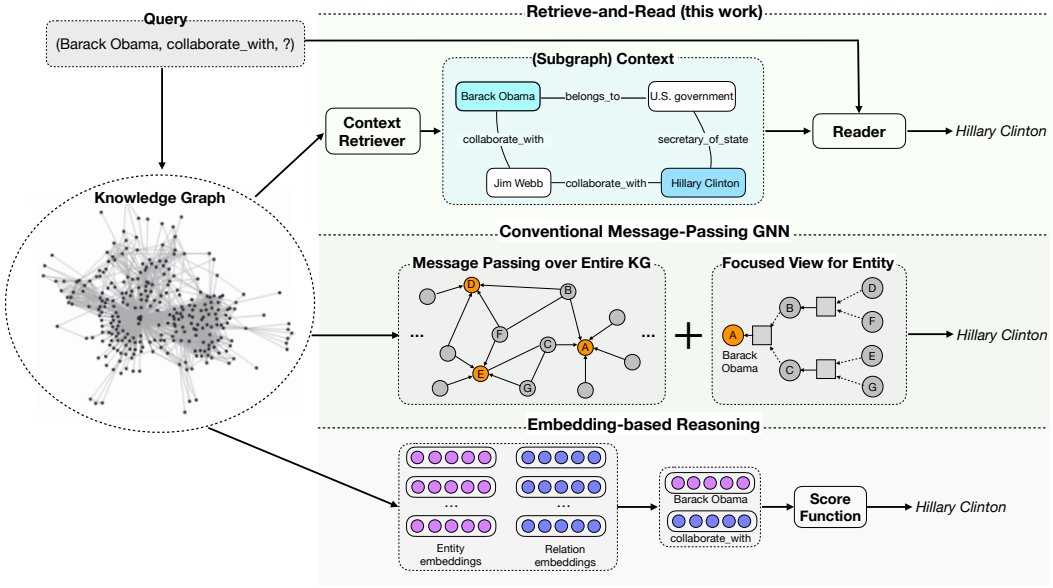


Figure 1: Overview of the proposed retrieve-and-read framework and comparison with existing frameworks for KG reasoning. Embedding-based methods try to encode all relevant information into the shallow embeddings, while message-passing graph neural networks (GNNs) iteratively learn the representations through message passing over the entire KG. In contrast, in our framework, we first retrieve a small context subgraph that is relevant to each input query, and jointly encode the query and the context for the final prediction. Here for simplicity we assume the context to be a connected subgraph, but being connected is not a necessary condition.

relevant subgraph from the KG as *context*, we can then easily use a high-capacity model to *read* the query in the corresponding context to make the final inference. To this end, we propose a novel *retrieve-and-read framework* for KG reasoning (see Figure 1 for an overview and comparison with existing frameworks). It consists of two main components: a *retriever* that identifies the relevant KG subgraph as context for the input query, and a *reader* that jointly considers the query and the retrieved context for inferring the answer. Such a retrieve-and-read framework has been widely used for the open-domain question answering (QA) problem (Chen et al., 2017; Zhu et al., 2021a), which faces a similar fundamental challenge: it also needs to answer a question in a massive corpus where only a small fraction is relevant to each specific question. The modularization provided by this framework has enabled rapid progress on retriever and reader models separately (Karpukhin et al., 2020; Xiong et al., 2021; Asai et al., 2020; Khattab et al., 2021; Glass et al., 2020; Deng et al., 2021).

Embracing the retrieve-and-read framework for KG reasoning could bring multiple potential advantages: 1) It provides great flexibility to explore and develop diverse models for retriever and reader separately. For example, we will explore several different choices for the retriever, even including some existing KG reasoning model. Because the reader only needs to deal with a small subgraph instead of the entire KG, we could easily use high-capacity models such as the Transformer (Vaswani et al., 2017), which has proven extremely successful for other tasks but the application on KG reasoning has been limited. 2) Relatedly, separate and more focused progress can be made on each component, which can then be combined to form new KG reasoning models. 3) Instead of learning and levering the same *static* representation for all inferences as in existing frameworks, the reader can dynamically learn a contextualized representation for each query and context for more accurate prediction. 4) Finally, this framework could potentially lead to scalable KG reasoning models for large-scale KGs, similar to how it has enabled open-domain QA to scale up to web-scale corpora (Karpukhin et al., 2020; Zhu et al., 2021a).

To demonstrate the effectiveness of the proposed framework, we propose a novel instantiation of the framework, KG-R3 (KG Reasoning with Retriever and Reader). It uses an existing KG reasoning method (Das et al., 2018) as retriever and a novel *Transformer-based GNN* as reader. Existing GNNs are mostly based on message passing (Gilmer et al., 2017), where the representation of a particular

node is iteratively updated by its neighbors. A known issue with message-passing GNNs is over-smoothing (Li et al., 2018), i.e., the representation of distinct nodes become indistinguishable as GNNs get deeper, which limits model capacity. The time complexity of message passing also grows exponentially with the number of layers, making it even harder to increase the capacity of such models. On the other hand, the Transformer model (Vaswani et al., 2017) has been driving the explosive growth of high-capacity models such as BERT (Devlin et al., 2018). The Transformer can support very high-capacity models (Brown et al., 2020; Chowdhery et al., 2022), which is one of the key reasons for its success. While it is challenging to apply Transformer to the entire KG due to its limited context window, the small context in the retrieve-and-read framework makes it feasible. We design a novel Transformer-based GNN which has a two-tower structure to separately encode the query and the context subgraph and a cross-attention mechanism to enable deep fusing of the two towers. A graph-induced attention structure is also developed to encode the context subgraph.

The major contribution of this work is three-fold:

- We propose a novel retrieve-and-read framework for knowledge graph reasoning.
- We develop a novel instantiation, KG-R3, of the framework, which consists of the first Transformer-based graph neural network for KG reasoning.
- We conduct empirical experiments on the standard FB15K-237 (Toutanova & Chen, 2015) and WN18RR (Dettmers et al., 2018) datasets and show that KG-R3 achieves competitive results with state-of-the-art methods.

2 Related Work

Graph Neural Networks. Graph Neural Networks have emerged as a popular class of neural networks for machine learning on graphs. Graphs naturally encode rich semantics of underlying data. Early models (Bruna et al., 2013; Kipf & Welling, 2016; Defferrard et al., 2016) extended the spectral convolution operation to graphs. Follow-up works (Battaglia et al., 2016; Veličković et al., 2017; Bresson & Laurent, 2017) introduced attention and gating mechanisms to aggregate the salient information from a node's neighborhood. These aforementioned models are applicable only to homogeneous graphs. In our present work, we develop a novel transformer-based GNN as the reader module for link prediction in multi-relational graphs like KGs.

KG reasoning models. KG reasoning has received significant attention in the research community in the past decade. The proposed models range from translation-based models (Bordes et al., 2013; Lin et al., 2015) to the ones that leverage convolutional neural networks (Dettmers et al., 2018; Nguyen et al., 2018). These shallow embedding methods learn embeddings for each entity/relation and use a parameterized score function to predict the plausibility of a "(subject, relation, object)" triple. To make use of rich graph neighborhood, several approaches have tried to adapt GNNs to multi-relational graphs for KG reasoning. Schlichtkrull et al. (2018) introduce relation-type dependent aggregation message aggregation. Teru et al. (2020) introduce a novel edge attention operation for aggregation in GNNs for the task of inductive relation prediction using the subgraph context. These methods use graph aggregation over the entire KG, thus limiting their application to large-scale KGs. In contrast, since we use subgraphs as the model input, our approach can potentially scale to large-scale KGs.

Path-based KB reasoning. Another line of work uses multi-hop paths to synthesize information for predicting the missing facts in a KG. DeepPath (Xiong et al., 2017) and MINERVA (Das et al., 2018) formulate it as sequential decision making problem and use reinforcement learning to search paths to the target entity. For our retriever module, we use MINERVA as one of the baseline methods. Yang et al. (2017) and Sadeghian et al. (2019) use inductive logic programming to assign weights to different paths for link prediction. Though these approaches are interpretable, they suffer from relatively poor performance compared to embedding based KG reasoning methods. Our proposed framework can utilize the subgraphs generated by these approaches for improved performance.

Transformers for Graph ML tasks. Transformers based models have seen widespread interest in the domain of graph ML. Dwivedi & Bresson (2021) adapt the GNN neighborhood aggregation to use Transformer-like self-attention. GRAPH-BERT (Zhang et al., 2020) uses the Transformer model for self-supervised learning of node representations. Graphormer (Ying et al., 2021) introduces inductive biases such as centrality encoding, spatial encoding and edge encoding to Transformer

model, leading to improved performance on OGB benchmarks (Hu et al., 2021). These approaches share the same limitation of aggregation over the whole graph, leading to poor scalability.

Open-domain Question Answering. The task of open-domain QA is to answer a question using knowledge from a massive corpus such as Wikipedia. A popular and successful way to address the challenge of large scale is through a two-stage retrieve-and-read pipeline (Chen et al., 2017; Zhu et al., 2021a), which leads to rapid developments of retriever and reader separately (Karpukhin et al., 2020; Xiong et al., 2021; Asai et al., 2020; Khattab et al., 2021; Glass et al., 2020; Deng et al., 2021). We draw inspiration from this pipeline and propose to use a retrieve-and-read framework for KG reasoning.

3 METHODOLOGY

Knowledge Graph. Given a set of entities $\mathcal{E}$ and a set of relations $\mathcal{R}$, a knowledge graph can be defined as a a collection of facts $\mathcal{F} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ where for each fact $f = (h, r, t),\ h, t \in \mathcal{E}, r \in \mathcal{R}$.

Link Prediction. The task of link prediction is to infer the missing facts in a KG. Given a link prediction query $(h, r, ?)$ or $(?, r, t)$, the model ranks the target entity among the set of candidate entities.

3.1 RETRIEVER

The function of the retriever module is to select a subset of the KG relevant to the query. This resultant subset is called a *subgraph*. We use the following off-the-shelf methods to generate subgraph inputs for the Transformer-based reader model in our framework:

- **Breadth-first search**: For breadth-first search, we sample edges starting from the source entity in breadth-first order till we reach the context budget.
- **One-hop neighborhood**: The one-hop neighborhood comprises of edges in the immediate one-hop neighborhood of the source entity.
- **MINERVA** (Das et al., 2018): MINERVA formulates KG reasoning as generation of multi-hop paths from the source entity to the target entity. The environment is represented as a Markov Decision Process on the KG, where the reinforcement learning agent gets a positive reward on reaching the target entity. The set of paths generated by MINERVA provides an interpretable provenance for KB reasoning. The retriever model utilizes the union of these paths decoded using beam search as the subgraph output.

Among these approaches, breadth-first search and one-hop neighborhood make use of uninformed search, i.e., they only enrich the query using surrounding context without going towards the target. On the other hand, the subgraph obtained using MINERVA aims to provide context which encloses information towards reaching the target entity.

3.2 READER ARCHITECTURE

Embedding layer. The input to the Transformer is obtained by summing the token lookup embedding, token type embedding and the segment embedding (Figure 2):

- **Token lookup embedding**: We use learned lookup embeddings for all entities and relations in the KG. These lookup embeddings store the global semantic information for each token.
- **Token type embedding**: Entities and relations have different semantics, so we use token type embeddings help the model distinguish between them.
- **Segment embedding**: It denotes whether a particular entity token corresponds to the terminal entity in a path starting from the source entity. This helps the model to differentiate between the terminal tokens, which are more likely to correspond to the final answer v.s. others.

The input to the model is query e.g. $(h, r, ?)$ and the associated subgraph g, a connected subset of the KG. The subgraph consists of nodes $\{e_1, e_2, \cdots, e_m\} \in \mathcal{E}$ and edges $\{r_1, r_2, \cdots, r_n\} \in \mathcal{R}$.

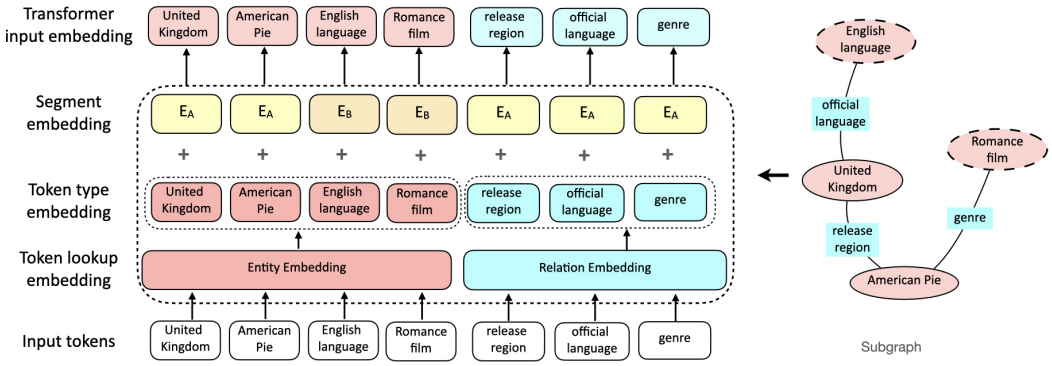


Figure 2: Schematic representation of embedding layer for subgraph input. The Transformer input is the sum of token lookup embedding, token type embedding and segment embedding.

The input sequence is constructed by concatenating the nodes and edges. Each edge has a unique token in the input though there might be multiple edges with the same predicate. The query and subgraph are first encoded by their respective Transformer encoders, which make use of graph-induced attention structure (details in below). Then the cross-attention module is used to modulate the subgraph representation, conditioned on the query.

Graph-induced Self-Attention. The attention structure ($\mathcal{A}_i$) governs the set of tokens that a particular token can attend to in the self-attention layer of the Transformer model, which aims to incorporate the (sub)graph structure into the transformer representations. We define the attention structure such that 1) all node tokens can attend to each other; 2) all edge tokens can attend to each other, and 3) for a particular triple (h, r, t), the token pairs (h, r) and (r, t) can attend to each other. This design is motivated by the need to balance the immediate graph neighborhood of a token v.s. its global context in the subgraph.

More formally, let $\{h_i^\ell\}_{i=1}^{m+n}$ denote the hidden representations of the tokens in layer ℓ.

$$h_i^{\ell+1} = O_h^\ell \Big\|_{k=1}^{H} \Big(\sum_{j \in \mathcal{A}_i} w_{ij}^{k,\ell} V^{k,\ell} \boldsymbol{h}_j^\ell \Big) \tag{1}$$

$$w_{ij}^{k,\ell} = \text{softmax}_{j \in \mathcal{A}_i} \Big(\frac{Q^{k,\ell} \boldsymbol{h}_i^\ell \cdot K^{k,\ell} \boldsymbol{h}_j^\ell}{\sqrt{d_k}} \Big) \tag{2}$$

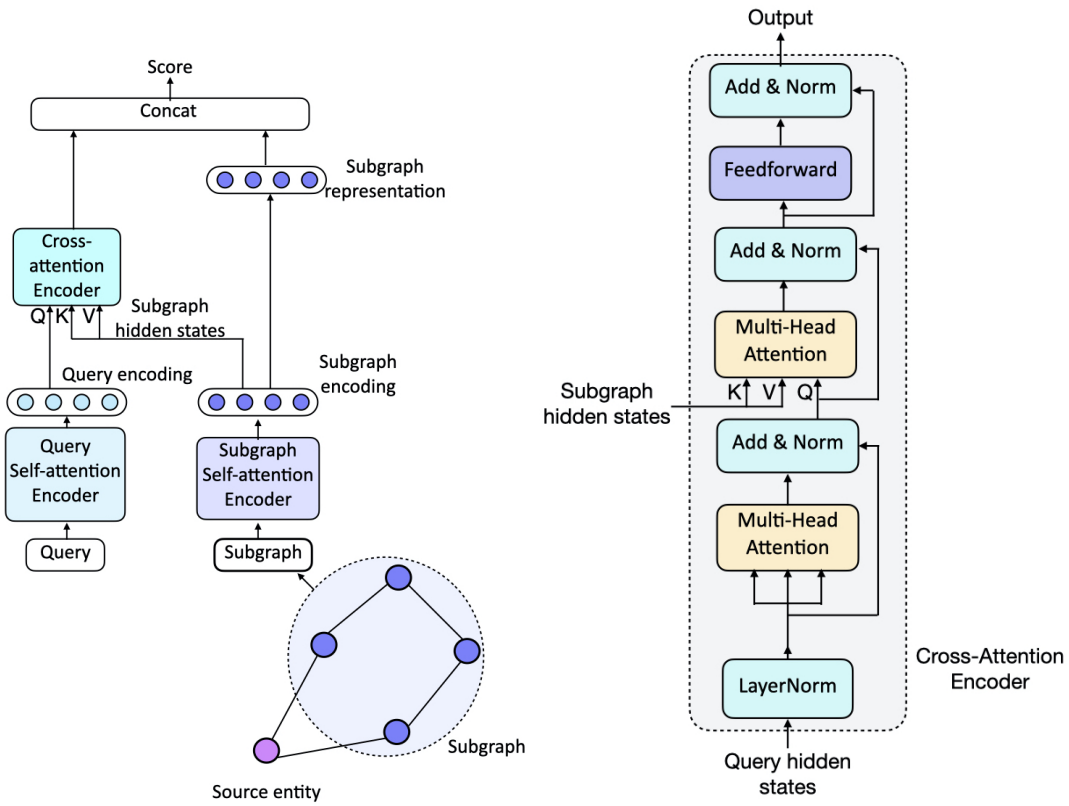
Here, $Q^{k,\ell}, K^{k,\ell}, V^{k,\ell} \in \mathbb{R}^{d_k \times d}$, $O_h^\ell \in \mathbb{R}^{d \times d}$ are projection matrices, H denotes the number of attention heads, d_k denotes the hidden dim. of keys and $\|$ denote concatenation.

Cross-Attention. In order to answer a link prediction query, the model needs a way to filter the subset of the edges in the subgraph relevant for a particular link prediction query. To accomplish this, we introduce cross-attention from the query to the subgraph (Figure 3b). Following Vaswani et al. (2017), the queries come from query hidden states whereas the keys and values are provided by the subgraph hidden states. The resultant representation encodes the subgraph information relevant to the query at hand. This is concatenated with the contextualized representation of the source entity in the subgraph to output the feature vector for predicting the plausibility scores. Figure 3a illustrates the overall model architecture for the Transformer-based reader.

More formally, Let $\{\boldsymbol{e}_i^{\text{q},\ell}\}$ and $\{\boldsymbol{e}_i^{\text{sub}}\}$ denote the self-attention hidden representations of query and subgraph respectively.

$$\text{Cross-Attention}(\{\boldsymbol{e}_i^{\text{q},\ell}\}, \{\boldsymbol{e}_i^{\text{sub}}\}) = O_h^\ell \Big\|_{k=1}^{H} \Big(\sum_{j=1}^{m+n} w_{ij}^{k,\ell} V^{k,\ell} \boldsymbol{e}_j^{\text{sub}} \Big) \tag{3}$$

$$w_{ij}^{k,\ell} = \text{softmax}\Big(\frac{Q^{k,\ell} \boldsymbol{e}_i^{\text{q},\ell} \cdot K^{k,\ell} \boldsymbol{e}_j^{\text{sub}}}{\sqrt{d_k}} \Big) \tag{4}$$



(a) Overall model architecture

(b) Cross-Attention Encoder architecture.

Figure 3: Reader module architecture. In the cross-attention encoder, the link prediction query serves as query input, while the subgraph serves as key and value input.

For a link prediction query, e.g., $(h, r, ?)$ the model predicts a score distribution over all tail entities. The model is trained using cross-entropy loss, framing it as a multi-class classification problem.

4 Experiments

4.1 Datasets

We use standard link prediction benchmarks FB15K-237 (Toutanova & Chen, 2015) and WN18RR (Dettmers et al., 2018) to evaluate our model. FB15K-237 is a subset of the original FB15K dataset after removing the train-test leakage due to inverse triplets. Similarly, WN18RR is a subset of Wordnet (Fellbaum, 2010) which is a lexical knowledge base. The statistics of these two datasets are given in Table 1. Among these, WN18RR is much sparser than FB15K-237.

4.2 Evaluation Protocol

For each test triplet (h, r, t), we corrupt either the head (corresponding to the link prediction query $(?, r, t)$) or the tail entity and rank the correct entity among all entities in the KG. Following (Bordes et al., 2013), we use the filtered evaluation setting i.e. the rank of a target entity is not affected by alternate correct entities. We report results on standard evaluation metrics: Mean Reciprocal Rank (MRR), Hits@1, Hits@3 and Hits@10.

4.3 Implementation Details

We implement our models in Pytorch (Paszke et al., 2017). We use $L = 3, A = 8, H = 320$ for the Transformer model (both self-attention and cross-attention), where L, A and H denote the number

of layers, number of attention heads per layer and the hidden size respectively. We use the Adamax (Kingma & Ba, 2015) optimizer for training. The learning rate schedule includes warmup for 10% of the training steps followed by linear decay. For both datasets, we tune the learning rate on the development set and report results on the test set with the best development setting. The batch size is set to 512. For MINERVA retriever, we use decoding beam size of 100 and 40 for FB15K-237 and WN18RR respectively. For BFS retriever, we use upto 100 and 30 edges for FB15K-237 and WN18RR respectively. Following (Chen et al., 2020), the one-hop neighborhood retriever uses 50 and 12 edges for FB15K-237 and WN18RR respectively.

Dataset	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{R}\rvert$	# Facts			Avg. node degree
			Train	Valid	Test	
FB15K-237	14,505	237	272,115	17,535	20,466	18.76
WN18RR	40,945	11	86,835	3,034	3,134	2.14

Table 1: Dataset statistics

		FB15K-237				WN18RR			
Framework	**Model**	**MRR**	**H@1**	**H@3**	**H@10**	**MRR**	**H@1**	**H@3**	**H@10**
Embedding-based	RESCAL (Nickel et al., 2011)	.356	.266	.390	.535	.467	.439	.478	.516
	TransE (Bordes et al., 2013)	.310	.218	.345	.495	.232	.061	.366	.522
	DistMult (Yang et al., 2014)	.342	.249	.378	.531	.451	.414	.466	.523
	ComplEx (Trouillon et al., 2016)	.343	.250	.377	.532	.479	.441	.495	.552
	ComplEx-N3 (Lacroix et al., 2018)	.350	-	-	.540	.470	-	-	.540
	RotatE (Sun et al., 2019)	.338	.241	.375	.533	.476	.428	.492	.571
CNN-based	ConvKB (Nguyen et al., 2018)	.243	.155	.371	.421	.249	.057	.417	.524
	ConvE (Dettmers et al., 2018)	.338	.247	.372	.521	.439	.409	.452	.499
Path-based	NeuralLP (Yang et al., 2017)	.240	-	-	.362	.435	.371	.434	.566
	DRUM (Sadeghian et al., 2019)	.343	.255	.378	.516	.486	.425	.513	.586
GNN-based	R-GCN (Schlichtkrull et al., 2018)	.248	.151	-	.417	-	-	-	-
	CompGCN (Vashishth et al., 2020)	.355	.264	.390	.535	.479	.443	.494	.546
	NBFNet (Zhu et al., 2021b)	.415	.321	.454	.599	.551	.497	.573	.666
Transformer-based	KG-BERT (Yao et al., 2019)	-	-	-	.420	-	-	-	.524
	HittER (Chen et al., 2021)	.373	.279	.409	.558	.503	.462	.516	.584
	KG-R3 (this work)	.387	.313	.408	.536	.458	.421	.471	.532

Table 2: Comparison of our framework with baseline methods. For all metrics, higher is better. Missing values are denoted by -. The baseline metrics correspond to best results obtained after extensive hyper-parameter tuning (Ruffinelli et al., 2020).

4.4 Main Results

Table 2 shows the overall link prediction results. We compare our model with several translation-based methods: TransE (Bordes et al., 2013), RESCAL (Nickel et al., 2011), DistMult (Yang et al., 2014), RotatE (Sun et al., 2019), ComplEx (Trouillon et al., 2016), CNN-based methods: ConvE (Dettmers et al., 2018), ConvKB (Nguyen et al., 2018), Path-based KB reasoning methods: NeuralLP (Yang et al., 2017), DRUM (Sadeghian et al., 2019), message-passing GNN methods: R-GCN (Schlichtkrull et al., 2018), CompGCN (Vashishth et al., 2020), NBFNet (Zhu et al., 2021b) and Transformer-based methods: KG-BERT (Yao et al., 2019) and HittER (Chen et al., 2021). We don't include baselines that use extra information such as description information for comparison. Also, we omit path-based methods that report link prediction performance only for one direction.

For FB15K-237, our proposed model with MINERVA subgraphs outperforms all embedding-based baselines, CNN-based approaches, path-based approaches, and HittER (Chen et al., 2020), a baseline Transformer model. This shows that the inductive biases in our model help it better utilize neighboring context. For WN18RR dataset, our model is competitive compared to established baselines.

4.5 FINE-GRAINED ANALYSIS

To gain further insights into the reader, we report a breakdown of the link prediction performance based on whether the target entity is present in the input subgraph (Table 3). When the target entity is present in the subgraph, the performance is very high (Hits@1 is almost $8\times$ the value when it is absent). This can be explained by the fact that the coverage of target entity provides the reader some potentially correct reasoning paths to better establish the link between the source and target entity. This also shows that the coverage of the target entity in the subgraph could be a useful indicator of the retriever module's performance.

	MRR	Hits@1	Hits@3	Hits@10
Target entity present in subgraph	**.675**	**.591**	**.714**	**.846**
Target entity absent in subgraph	.135	.073	.139	.259

Table 3: Performance breakdown based on whether the target entity is present in the input subgraph on FB15K-37 dev. set. The performance is signicantly better when the target entity is present in the subgraph.

	FB15K-237				**WN18RR**			
Model	**MRR**	**H@1**	**H@3**	**H@10**	**MRR**	**H@1**	**H@3**	**H@10**
MINERVA	**.390**	**.317**	**.411**	**.541**	**.472**	**.435**	**.486**	**.542**
BFS	.303	.215	.334	.475	.370	.308	.412	.479
One-hop neigh.	.303	.215	.330	.481	.437	.395	.456	.516

Table 4: Ablations for *Retriever* module (FB15K-237 dev. set). MINERVA retriever outperforms others by a wide margin.

4.6 ABLATION STUDIES

We do several ablations of both the retriever and the reader modules to understand the contribution of different components towards the final performance. For the retriever, we experiment with three choices - MINERVA, breadth-first search and one-hop neighborhood (Table 4). For both datasets, the MINERVA retriever outperforms BFS and one-hop neighborhood by a significant margin. This can be attributed to the fact that MINERVA is explicitly trained to find paths that lead to target entity using RL, whereas the other two approaches correspond to uninformed search strategies. This provides interesting insights into development of a good retriever. We further analyse the statistics for target entity coverage in the subgraph (Table 5). This shows that higher target entity coverage in subgraph potentially leads to better performance.

Retriever	Target entity coverage (%)
MINERVA	**46.28**
BFS	16.44

Table 5: Comparison of target entity coverage for different retriever methods (FB15K-237 dev. set). MINERVA has much better coverage than BFS.

For the reader, we experiment with omitting the cross-attention layers, omitting the subgraph representation in the aggregate representation for link prediction, omitting the query representation and using fully-connected attention in Transformer instead of the graph-induced attention structure. The results are shown in Table 6. The most signficant drop in performance is caused by dropping the graph-induced attention structure, which shows that our novel attention design plays a key role in overall performance. The use of cross-attention brings notable improvement in metrics such as Hits@1. Among the query and subgraph feature representation, the former has a greater contribution to the the performance.

Model	MRR	Hits@1	Hits@3	Hits@10
Ours	**.390**	**.317**	.411	.541
– Cross Attention	.387	.309	**.414**	.544
– Graph Attention structure	.326	.240	.353	.497
– Subgraph embed	.370	.277	.406	**.556**
– Query embed	.359	.277	.384	.527

Table 6: Ablations for *Reader* module (FB15K-237 dev. set). Graph Attention structure contributes the most towards final performance.

4.7 Efficiency analysis

Table 7 shows the comparison of training and inference complexity (per triplet) of our method to two prominent GNN baselines - R-GCN and NBFNet. **Note**: The calculation includes the complexity of MINERVA retriever $O(d^2 + d\frac{|\mathcal{E}|}{|\mathcal{V}|})$.

Model	Training complexity	Inference complexity (per triplet)
R-GCN (Schlichtkrull et al., 2018)	$O(T(\|\mathcal{E}\|d^2))$	$O(\|\mathcal{E}\|d^2)$
NBFNet (Zhu et al., 2021b)	$O(T(\|\mathcal{E}\|d + \|\mathcal{V}\|d^2))$	$O(\|\mathcal{E}\|d + \|\mathcal{V}\|d^2)$
Ours	$O(T((n+e)^2 d + (n+e)d^2 + d^2 + d\frac{\|\mathcal{E}\|}{\mathcal{V}}))$	$O((n+e)^2 d + (n+e)d^2 + d^2 + d\frac{\|\mathcal{E}\|}{\|\mathcal{V}\|})$

Table 7: Comparison of complexity of our approach to baseline methods. Here, n and e denote the average number of nodes and edges in a subgraph respectively. $|\mathcal{E}|$ and $|\mathcal{V}|$ denote the number of edges and nodes in the KG respectively. T is the no. of iterations needed for convergence and d is hidden dimension.

Since $(n + e) << |\mathcal{E}|$, our approach is clearly more efficient than R-GCN both for training and inference. For comparison with NBFNet, the size of subgraph $(n + e)$ is much smaller than the total number of nodes $|\mathcal{V}|$ in the KG and $(n + e)^2 < |\mathcal{E}|$, so our method is potentially more efficient than NBFNet for large-scale KGs.

5 Discussion and Conclusion

In this work, we propose a retrieve-and-read framework for knowledge graph reasoning. We develop a novel instantiation, KG-R3, of the framework, which consists of the first Transformer-based graph neural network for KG reasoning. While being an initial exploration of our proposal, empirical experiments on standard benchmarks show that KG-R3 achieves competitive results with state-of-the-art methods, which indicates great potential of the proposed framework.

One of the drawbacks of our proposed framework is, the two-stage pipeline design may lead to cascading errors in cases when the subgraph is sub-optimal. One promising direction to improve this aspect is to model the subgraph as a latent variable, where the retriever and reader can function in synergy to mutually enhance each other. Standard latent variable-based approaches such as expectation-maximization (EM) can be used to give feedback from the reader module to the retriever s.t. it can learn to retrieve subgraphs which better help the reader to infer the target, and vice versa. We leave these investigations as future work. As another future research direction, we would like to make the reader more robust to noisy subgraphs and explore other baselines for the retriever.

We believe this new framework will be a useful resource for the research community to accelerate the development of high-performance and scalable graph-based models for KG reasoning. Future work will involve deploying this framework for link prediction on large KGs. We would also like to explore its applications in other KG tasks.

REFERENCES

Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. Learning to retrieve reasoning paths over wikipedia graph for question answering. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJgVHkrYDH.

Peter W Battaglia, Razvan Pascanu, Matthew Lai, Danilo Rezende, and Koray Kavukcuoglu. Interaction networks for learning about objects, relations and physics. *arXiv preprint arXiv:1612.00222*, 2016.

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

Antoine Bordes, Sumit Chopra, and Jason Weston. Question answering with subgraph embeddings. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 615–620, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1067. URL https://aclanthology.org/D14-1067.

Xavier Bresson and Thomas Laurent. Residual gated graph convnets. *arXiv preprint arXiv:1711.07553*, 2017.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

Pablo Castells, Miriam Fernandez, and David Vallet. An adaptation of the vector-space model for ontology-based information retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 19(2):261–272, 2007. doi: 10.1109/TKDE.2007.22.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1870–1879, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1171. URL https://aclanthology.org/P17-1171.

Sanxing Chen, Xiaodong Liu, Jianfeng Gao, Jian Jiao, Ruofei Zhang, and Yangfeng Ji. Hitter: Hierarchical transformers for knowledge graph embeddings. *arXiv preprint arXiv:2008.12813*, 2020.

Sanxing Chen, Xiaodong Liu, Jianfeng Gao, Jian Jiao, Ruofei Zhang, and Yangfeng Ji. HittER: Hierarchical transformers for knowledge graph embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 10395–10407, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.812.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, and Andrew McCallum. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Syg-YfWCW.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29:3844–3852, 2016.

Xiang Deng, Yu Su, Alyssa Lees, You Wu, Cong Yu, and Huan Sun. ReasonBERT: Pre-trained to reason with distant supervision. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6112–6127, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.494. URL https://aclanthology.org/2021.emnlp-main.494.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *AAAI Workshop on Deep Learning on Graphs: Methods and Applications*, 2021.

Christiane Fellbaum. Wordnet. In *Theory and applications of ontology: computer applications*, pp. 231–243. Springer, 2010.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pp. 1263–1272. PMLR, 2017.

Michael Glass, Alfio Gliozzo, Rishav Chakravarti, Anthony Ferritto, Lin Pan, G P Shrivatsa Bhargav, Dinesh Garg, and Avi Sil. Span selection pre-training for question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2773–2782, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.247. URL https://aclanthology.org/2020.acl-main.247.

Weihua Hu, Matthias Fey, Hongyu Ren, Maho Nakata, Yuxiao Dong, and Jure Leskovec. OGB-LSC: A large-scale challenge for machine learning on graphs. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=qkcLxoC52kL.

Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 687–696, 2015.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.550. URL https://aclanthology.org/2020.emnlp-main.550.

Omar Khattab, Christopher Potts, and Matei Zaharia. Baleen: Robust multi-hop reasoning at scale via condensed retrieval. *Advances in Neural Information Processing Systems*, 34, 2021.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. Canonical tensor decomposition for knowledge base completion. In *ICML*, 2018.

Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Thirty-Second AAAI conference on artificial intelligence*, 2018.

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

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung. A novel embedding model for knowledge base completion based on convolutional neural network. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 327–333, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2053. URL https://www.aclweb.org/anthology/N18-2053.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *Icml*, volume 11, pp. 809–816, 2011.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

Daniel Ruffinelli, Samuel Broscheit, and Rainer Gemulla. You can teach an old dog new tricks! on training knowledge graph embeddings. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BkxSmlBFvr.

Ali Sadeghian, Mohammadreza Armandpour, Patrick Ding, and Daisy Zhe Wang. Drum: End-to-end differentiable rule mining on knowledge graphs. *Advances in Neural Information Processing Systems*, 32, 2019.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pp. 593–607. Springer, 2018.

Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. End-to-end structure-aware convolutional networks for knowledge base completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3060–3067, 2019.

Wei Shen, Jianyong Wang, and Jiawei Han. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):443–460, 2015. doi: 10.1109/TKDE.2014.2327028.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. In *Advances in neural information processing systems*, pp. 926–934, 2013.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HkgEQnRqYQ.

Komal Teru, Etienne Denis, and Will Hamilton. Inductive relation prediction by subgraph reasoning. In *International Conference on Machine Learning*, pp. 9448–9457. PMLR, 2020.

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*, pp. 57–66, 2015.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. International Conference on Machine Learning (ICML), 2016.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. Composition-based multi-relational graph convolutional networks. *arXiv preprint arXiv:1911.03082*, 2019.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. Composition-based multi-relational graph convolutional networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BylA_C4tPr.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

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

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph and text jointly embedding. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1591–1601, 2014.

Jason Weston, Antoine Bordes, Oksana Yakhnenko, and Nicolas Usunier. Connecting language and knowledge bases with embedding models for relation extraction. *arXiv preprint arXiv:1307.7973*, 2013.

Wenhan Xiong, Thien Hoang, and William Yang Wang. DeepPath: A reinforcement learning method for knowledge graph reasoning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 564–573, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1060. URL https://aclanthology.org/D17-1060.

Wenhan Xiong, Xiang Li, Srini Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Scott Yih, Sebastian Riedel, Douwe Kiela, and Barlas Oguz. Answering complex open-domain questions with multi-hop dense retrieval. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=EMHoBG0avc1.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*, 2014.

Fan Yang, Zhilin Yang, and William W Cohen. Differentiable learning of logical rules for knowledge base reasoning. *Advances in neural information processing systems*, 30, 2017.

Liang Yao, Chengsheng Mao, and Yuan Luo. Kg-bert: Bert for knowledge graph completion. *ArXiv*, abs/1909.03193, 2019.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=OeWooOxFwDa.

Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 353–362, 2016.

Jiawei Zhang, Haopeng Zhang, Congying Xia, and Li Sun. Graph-bert: Only attention is needed for learning graph representations. *arXiv preprint arXiv:2001.05140*, 2020.

Shuangjia Zheng, Jiahua Rao, Ying Song, Jixian Zhang, Xianglu Xiao, Evandro Fei Fang, Yuedong Yang, and Zhangming Niu. Pharmkg: a dedicated knowledge graph benchmark for bomedical data mining. *Briefings in bioinformatics*, 22(4):bbaa344, 2021.

Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. Retrieving and reading: A comprehensive survey on open-domain question answering. *arXiv preprint arXiv:2101.00774*, 2021a.

Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhonneux, and Jian Tang. Neural bellman-ford networks: A general graph neural network framework for link prediction. *Advances in Neural Information Processing Systems*, 34:29476–29490, 2021b.