HittER: Hierarchical Transformers for Knowledge Graph Embeddings

Sanxing Chen[*]
University of Virginia
sc3hn@virginia.edu

Xiaodong Liu, Jianfeng Gao
Microsoft Research
{xiaodl,jfgao}@microsoft.com

Jian Jiao, Ruofei Zhang
Microsoft Bing Ads
{jiajia,bzhang}@microsoft.com

Yangfeng Ji
University of Virginia
yangfeng@virginia.edu

Abstract

This paper examines the challenging problem of learning representations of entities and relations in a complex multi-relational knowledge graph. We propose **HittER**, a **Hi**erarchical **T**ransformer model **t**o jointly learn **E**ntity-relation composition and **R**elational contextualization based on a source entity's neighborhood. Our proposed model consists of two different Transformer blocks: the bottom block extracts features of each entity-relation pair in the local neighborhood of the source entity and the top block aggregates the relational information from outputs of the bottom block. We further design a masked entity prediction task to balance information from the relational context and the source entity itself. Experimental results show that HittER achieves new state-of-the-art results on multiple link prediction datasets. We additionally propose a simple approach to integrate HittER into BERT and demonstrate its effectiveness on two Freebase factoid question answering datasets.

1 Introduction

Knowledge graphs (KG) are a major form of knowledge bases where knowledge is stored as graph-structured data. Because of their broad applications in various intelligent systems including natural language understanding (Logan et al., 2019; Zhang et al., 2019b; Hayashi et al., 2020) and reasoning (Riedel et al., 2013; Xiong et al., 2017; Bauer et al., 2018; Verga et al., 2021), learning representations of knowledge graphs has been studied in a large body of literature.

To learn high quality representations of knowledge graphs, many researchers adopt the idea of mapping the entities and relations in a knowledge graph to points in a vector space. These knowledge graph embedding (KGE) methods usually leverage geometric properties in the vector space, such as translation (Bordes et al., 2013), bilinear transformations (Yang et al., 2015, DistMult), or rotation (Sun et al., 2018). Multi-layer convolutional networks are also used for KGE (Dettmers et al., 2018, ConvE). Such KGE methods are conceptually simple and can be applied to tasks like factoid question answering (Saxena et al., 2020) and language modeling (Peters et al., 2019).

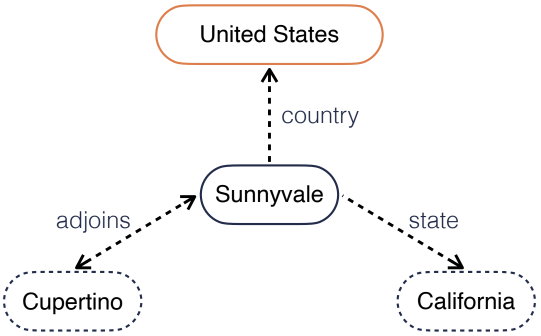

Figure 1: An example subgraph sampled from FB15K-237. Four nodes (entities) are connected by three different types of relations representing facts like `Sunnyvale` belongs to the state of `California`.

However, it is rather challenging to encode all of the information about an entity into a single vector. For example, to infer the missing object in the incomplete triplet `<Sunnyvale, county, ?>` (Figure 1), traditional KGE methods rely on the geographic information stored in the embedding of `Sunnyvale`. While we can read such information from its graph context, e.g., from a neighbor node that represents the state it belongs to (i.e., `California`). In this way, we allow the model to store and utilize information about an entity via its relational context. To implement this process, previous work uses graph neural networks (GNN) or attention-based approaches to learn representations based on both entities and their graph context (Kipf and Welling, 2017; Bansal et al., 2019;

[*] Work was done during an internship at Microsoft Bing Ads.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 10395–10407
November 7–11, 2021. ©2021 Association for Computational Linguistics

Vashishth et al., 2020). However, these methods are usually restricted in expressiveness because of the shallow network architecture they use.[1]

In this paper, we present HittER, a deep hierarchical Transformer model to learn representations of entities and relations in a knowledge graph jointly by aggregating information from graph neighborhoods. Although prior work shows Transformers can learn relational knowledge from large amounts of unstructured textual data (Jiang et al., 2020; Manning et al., 2020), HittER *explicitly operates over structured inputs using a hierarchical architecture*. Essentially, HittER consists of two levels of Transformer blocks. As shown in Figure 2, the bottom block provides relation-dependent entity embeddings for the neighborhood around an entity and the top block aggregates information from its graph context. To ensure HittER work across graphs of different properties, we further design a masked entity prediction task to balance the contextual relational information and information from the training entity itself.

We evaluate the proposed method using the link prediction task, which is one of the canonical tasks in statistical relational learning (SRL). Link prediction (essentially KG completion) serves as a good proxy to evaluate the effectiveness of learned graph representations, by measuring the ability of a model to generalize relational knowledge stored in training graphs to unseen facts. Meanwhile, it has an important application to knowledge graph completion given the fact that most of the knowledge graphs are still highly incomplete (West et al., 2014). Our approach achieves new state-of-the-art results on two standard benchmark datasets: FB15K-237 (Toutanova and Chen, 2015) and WN18RR (Dettmers et al., 2018).

Unlike the previous shallow KGE methods that cannot be trivially utilized by widely used Transformer-based models for language tasks (Peters et al., 2019), our approach benefits from the unified Transformer architecture and its extensibility. As a case study, we show how to integrate the learned representations of HittER into pre-trained language models like BERT (Devlin et al., 2019). Our experiments demonstrate that HittER significantly improves BERT on two Freebase factoid question answering (QA) datasets: FreebaseQA (Jiang et al., 2019) and WebquestionSP (Yih et al., 2016).

Our experimental code as well as multiple pre-trained models are publicly available.[2]

2 HittER

We introduce our proposed hierarchical Transformer model (Figure 2) in this section. In Section 2.1, we provide the background about how link prediction can be done with a simple Transformer scoring function. We then describe the detailed architecture of our proposed model in Section 2.2. Finally, we discuss our strategies to learn balanced contextual representations of an entity in Section 2.3.

2.1 Transformers for Link Prediction

A knowledge graph can be viewed as a set of triplets ($G = \{(e_s, r_p, e_o)\}$) and each has three items including the subject $e_s \in \mathcal{E}$, the predicate $r_p \in \mathcal{R}$, and the object $e_o \in \mathcal{E}$ to describe a single fact (link) in the knowledge graph. Our model approximates a pointwise scoring function $\psi : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \mapsto \mathbb{R}$ which takes a triplet as input and produces a score reflecting the plausibility of such fact triplet existing in the knowledge graph. In the task of link prediction, given a triplet with either the subject or the object missing, the goal is to find the missing entity from the set of all entities $\mathcal{E}$. Without loss of generality, we describe the case where an incomplete triplet (e_s, r_p) is given and we want to predict the object e_o. And vice versa, the subject e_s can be predicted in a similar process, except that a reciprocal predicate $r_{\tilde{p}}$ will be used to distinguish these two cases (Lacroix et al., 2018). We call the entity in the incomplete triplet the source entity e_{src} and call the entity we want to predict the target entity e_{tgt}.

Link prediction can be done in a straightforward manner with a Transformer encoder (Vaswani et al., 2017) as the scoring function, depicted inside the dashed box in Figure 2. Our inputs to the Transformer encoder are randomly initialized embeddings of the source entity e_{src}, the predicate r_p, and a special `[CLS]` token. Three different learned type embeddings are directly added to the three token embeddings similar to the input representations of BERT (Devlin et al., 2019). Then we use the output embedding corresponding to the `[CLS]` token ($M_{e_{src}}$) to predict the target entity, which is

[1] GNN methods' depth is tied to their receptive fields and thus constrained by over-smoothing issues (Liu et al., 2020).

[2] https://github.com/sanxing-chen/HittER

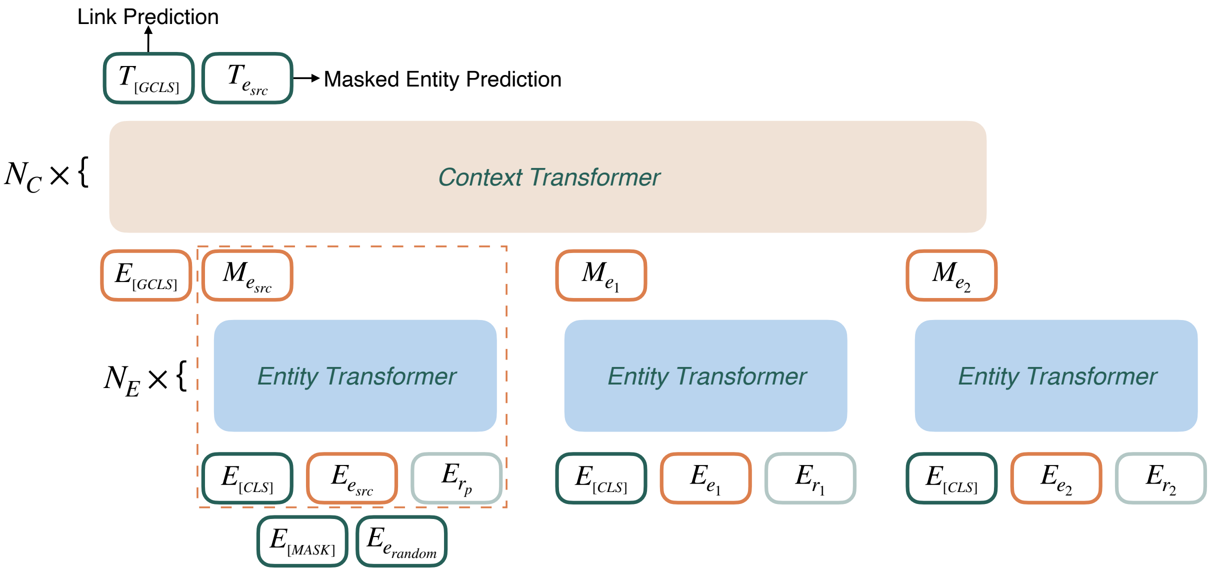

Figure 2: Our model consists of two Transformer blocks organized in a hierarchical fashion. The bottom Transformer block captures the interactions between a entity-relation pair while the top one gathers information from an entity's graph neighborhood. Taking the entity embeddings E_e and the relation embeddings E_r as input, the output embedding $T_{\text{[GCLS]}}$ is used for predicting the target entity. We sometimes mask or replace $E_{e_{src}}$ with $E_{[MASK]}$ or $E_{e_{random}}$. In which case, an additional output embedding $T_{e_{src}}$ can be used to recover the perturbed entity. The dashed box indicates a simple context-independent baseline where $M_{e_{src}}$ is directly used for link prediction.

implemented as follows. We first compute the plausibility score of the true triplet as the dot-product between $M_{e_{src}}$ and the token embedding of the target entity. In the same way, we also compute the plausibility scores for all other candidate entities and normalize them using the softmax function. Lastly, we use the normalized distribution to get the cross-entropy loss $\mathcal{L}_{\text{LP}} = -\log p(e_{tgt} \mid M_{e_{src}})$ for training. We will use this model as a simple *context-independent* baseline later in experiments, which is similar to the approach explored in Wang et al. (2019).

Although such simple Transformer encoder does a decent work in link prediction tasks, learning knowledge graph embeddings from one triplet at a time ignores the abundant structural information in the graph context. Our model, as described in the following section, also considers the relational neighborhood of the source vertex (entity), which includes all of its adjacent vertices in the graph, denoted as $N_G(e_{src}) = \{(e_{src}, r_i, e_i)\}$.[3]

2.2 Hierarchical Transformers

We propose a hierarchical Transformer model for knowledge graph embeddings (Figure 2). The proposed model consists of two blocks of multi-layer bidirectional Transformer encoders.

We employ the Transformer described in Section 2.1 as our bottom Transformer block, called the *entity Transformer*, to learn interactions between an entity and its associated relation type. Different from the context-independent scenario described in the last section, this entity Transformer is now generalized to also encode information from a relational context. In specific, there are two cases in our context-dependent scenario:

1. We consider the source entity with the predicate in the incomplete triplet as the first entity-relation pair;

2. We consider an entity from the graph neighborhood of the source entity with the relation type of the edge that connects them.

The bottom block is responsible of packing all useful features from the entity-relation pairs into vector representations to be further used by the top block. Compared with directly feeding all entity-relation pairs to the top block, it helps reduce the run-time of the model by converting two inputs to one.[4]

The top Transformer block is called the *context Transformer*. Given the output of the previous en-

[3] Our referred neighborhood is slightly different from the formal definition since we only consider edges connecting to the source vertex.

[4] This avoids long input sequences for Transformer's $\mathcal{O}(n^2)$ computation.

tity Transformer and a special `[GCLS]` embedding, it contextualizes the source entity with relational information from its graph neighborhood. Similarly, three type embeddings are assigned to the special `[GCLS]` token embedding, the intermediate source entity embedding, and the other intermediate neighbor entity embeddings. The cross-entropy loss for link prediction is now changed as follows.

$$\mathcal{L}_{\text{LP}} = -\log p(e_{tgt} \mid T_{[GCLS]}) \tag{1}$$

The top block does most of the heavy lifting to aggregate contextual information together with the information from the source entity and the predicate, by using structural features extracted from the output vector representations of the bottom block.

2.3 Balanced Contextualization

Our hierarchical Transformer model shows a simple way to introduce graph context to link prediction, however, trivially providing contextual information to the model could cause problems. On one hand, since a source entity often contains high-quality information for link prediction and learning to extract useful information from a broad noisy context requires substantial effort, the model could simply learn to ignore the additional contextual information. On the other hand, the introduction of rich contextual information could in turn downgrade information from the source entity and contain spurious correlations, which potentially lead to over-fitting based on our observation. To address these challenges, inspired by the successful Masked Language Modeling pre-training task in BERT, we propose a two-step *Masked Entity Prediction* task (MEP) to balance the utilization of source entity and graph context during contextualization process.

To avoid the first problem, we apply a masking strategy to the source entity of each training example as follows. During training, we randomly select a proportion of training examples in a batch. With certain probabilities, we replace the input source entity with a special mask token `[MASK]`, a random chosen entity, or just leave it unchanged. The purpose of these perturbations is to introduce extra noise to the information from the source entity, thus forcing the model to learn contextual representations. The probability of each category is dataset-specific hyper-parameter: for example, we can mask out the source entity more frequently if its graph neighborhood is denser (in which case, the source entity can be easily replaced by the additional contextual information).

In terms of the second problem, we want to promote the model's awareness of the masked entity. Thus we train the model to recover the perturbed source entity based on the additional contextual information. To do this, we use the output embedding corresponding to the source entity $T_{e_{src}}$ to predict the correct source entity via a classification layer.[5] We can add the cross-entropy classification loss to the previous mentioned link prediction loss as an auxiliary loss, as follows.

$$\mathcal{L}_{\text{MEP}} = -\log p(e_{src} \mid T_{e_{src}}) \tag{2}$$

$$\mathcal{L} = \mathcal{L}_{\text{LP}} + \mathcal{L}_{\text{MEP}} \tag{3}$$

This step is important when solely relying on the contextual clues is insufficient to do link prediction, which means the information from the source entity needs to be emphasized. And it is otherwise unnecessary when there is high-quality contextual information. However, the first step of entity masking is always beneficial to the utilization of contextual information according to our observations. Thus we use dataset-specific configurations to strike a balance between these two sides.

In addition to the MEP task, we implement a uniform neighborhood sampling strategy where only a fraction of the entities in the graph neighborhood will appear in a training example. This sampling strategy acts like a data augmenter and similar to the edge dropout regularization in graph neural network methods (Rong et al., 2020). We also have to remove the ground truth target entity from the source entity's neighborhood during training. Otherwise, it will create a dramatic train-test mismatch because the ground truth target entity can always be found from the source entity's neighborhood during training while it can rarely be found during testing. The model will thus learn to naively select an entity from the neighborhood.

3 Link Prediction Experiments

We describe our link prediction experiments in this section. Section 3.1 introduces two standard benchmark datasets we used. We then describe our evaluation protocol in Section 3.2, and the detailed experimental setup in Section 3.3. Our proposed method are assessed both quantitatively and qualitatively in Section 3.4. Besides, several ablation

[5] We share the same weight matrix in the input embeddings layer and the linear transformation of this classification layer.

Model	FB15K-237					WN18RR				
	#Params	MRR↑	Hits↑			#Params	MRR↑	Hits↑		
			@1	@3	@10			@1	@3	@10
RESCAL (Nickel et al., 2011)	6M	.356	.266	.390	.535	6M	.467	.439	.478	.516
TransE (Bordes et al., 2013)	2M	.310	.218	.345	.495	21M	.232	.061	.366	.522
DistMult (Yang et al., 2015)	4M	.342	.249	.378	.531	21M	.451	.414	.466	.523
ComplEx (Trouillon et al., 2016)	4M	.343	.250	.377	.532	5M	.479	.441	.495	.552
ConvE (Dettmers et al., 2018)	9M	.338	.247	.372	.521	36M	.439	.409	.452	.499
RotatE (Sun et al., 2018)	15M	.338	.241	.375	.533	20M	.476	.428	.492	.571
CoKE (Wang et al., 2019)	10M	.364	.272	.400	.549	17M	.484	.450	.496	.553
TuckER (Balazevic et al., 2019)	-	.358	.266	.394	.544	-	.470	.443	.482	.526
CompGCN (Vashishth et al., 2020)	-	.355	.264	.390	.535	-	.479	.443	.494	.546
RotH (Chami et al., 2020)	8M	.344	.246	.380	.535	21M	.496	.449	.514	**.586**
HittER	16M	**.373**	**.279**	**.409**	**.558**	24M	**.503**	**.462**	**.516**	.584

Table 1: Comparison between the proposed method and baseline methods. Results of RotatE, CoKE, TuckER, CompGCN, and RotH are taken from their original papers. Numbers in **bold** represent the best results.

studies are presented in Section 3.5 to demonstrate the importance of balanced contextualization.

3.1 Datasets

We train and evaluate our proposed method on two standard benchmark datasets FB15K-237 (Toutanova and Chen, 2015) and WN18RR (Dettmers et al., 2018) for link prediction, following the standard train/test split.[6] FB15K-237 is a subset sampled from the Freebase (Bollacker et al., 2008) with trivial inverse links removed. It stored facts about topics in movies, actors, awards, etc. WN18RR is a subset of the WordNet (Miller, 1995) which contains structured knowledge of English lexicons. Statistics of these two datasets are shown in Table 2. Notably, WN18RR is much sparser than FB15k-237 which implies it has less structural information in the local neighborhood of an entity. This will affect our configurations of the masked entity prediction task consequently.

3.2 Evaluation Protocol

The task of link prediction in a knowledge graph is defined as an entity ranking task. Essentially, for each test triplet, we remove the subject or the object from it and let the model predict which is the most plausible answer among all possible entities. After scoring all entity candidates and sorting them by the computed scores, the rank of the ground truth target entity is used to further compute various ranking metrics such as mean reciprocal rank (MRR) and hits@k, $k \in \{1, 3, 10\}$. We report all of these ranking metrics under the filtered setting proposed in Bordes et al. (2013) where valid entities except the ground truth target entity are filtered out from the rank list.

Dataset	FB15K-237	WN18RR
#Entities	14,541	40,943
#Relations	237	11
#Triples	310,116	93,003
#Avg. degree	42.7	4.5

Table 2: Dataset statistics. The WN18RR dataset is significantly sparser than the FB15K-237 dataset.

3.3 Experimental Setup

We implement our proposed method in PyTorch (Paszke et al., 2019) under the LibKGE framework (Broscheit et al., 2020). To perform a fair comparison with some early baseline methods, we reproduce their results using hyper-parameter configurations from LibKGE.[7] All data and evaluation metrics can be found in LibKGE.

Our model consists of a three-layer entity Transformer and a six-layers context Transformer. Each

[6] We intentionally omit the original FB15K and WN18 datasets because of their known flaw in test-leakage (Toutanova and Chen, 2015).

[7] These configurations consider many recent training techniques and are found by extensive searches. Thus the results are generally much better then the original reported ones.

Transformer layer has eight heads. The dimension size of hidden states is 320 across all layers except that we use 1280 dimensions for the position-wise feed-forward networks inside Transformer layers suggested by Vaswani et al. (2017). We set the maximum numbers of uniformly sampled neighbor entities for every example in the FB15K-237 and WN18RR dataset to be 50 and 12 respectively. Such configurations are intended to ensure most examples (more than 85% of the cases in each dataset) can have access to its entire local neighborhood during inference. During training, we further randomly drop 30% of entities from these fixed-size sets in both datasets.

We train our models using Adamax (Kingma and Ba, 2015) with a learning rate of 0.01 and an L2 weight decay rate of 0.01. The learning rate linearly increases from 0 over the first ten percent of training steps, and linearly decreases through the rest of the steps. We apply dropout (Srivastava et al., 2014) with a probability $p = 0.1$ for all layers, except that $p = 0.6$ for the embedding layers. We apply label smoothing with a rate 0.1 to prevent the model from being over-confident during training. We train our models using a batch size of 512 for at most 500 epochs and employ early stopping based on MRR in the validation set.

When training our model with the masked entity prediction task, we use the following dataset-specific configurations based on validation MRR in few early trials:

- **WN18RR**: 50% of examples are subjected to this task. Among them, 60% of examples are masked out, the rest are split in a 3:7 ratio for replaced and unchanged ones.

- **FB15K-237**: 50% of examples in a batch are masked out. No replaced or unchanged ones. We do not include the auxiliary loss.

Training our full models takes 7 hours (WN18RR) and 37 hours (FB15K-237) on a NVIDIA Tesla V100 GPU.

3.4 Experimental Results

Table 1 shows that the results of HittER compared with baseline methods including some early methods and previous SOTA methods.[8] We outperform all previous work by a substantial margin across nearly all the metrics. Comparing to some previous methods which target some observed patterns of specific datasets, our proposed method is more general and is able to give more consistent improvements over the two standard datasets. For instance, the previous SOTA in WN18RR, RotH explicitly captures the hierarchical and logical patterns by hyperbolic embeddings. Comparing to it, our model performs better especially in the FB15K-237 dataset which has a more diverse set of relation types. On the other hand, our models have comparable numbers of parameters to baseline methods, since entity embeddings contribute to the majority of the parameters.

Contextualization	FB15K-237		WN18RR	
	MRR	H@10	MRR	H@10
Balanced	$\mathbf{37.5}_{\mathbf{(.1)}}$	$\mathbf{56.1}_{\mathbf{(.1)}}$	$\mathbf{50.0}_{\mathbf{(.4)}}$	$\mathbf{58.2}_{\mathbf{(.4)}}$
Unbalanced	$36.7_{(.2)}$	$55.4_{(.4)}$	$47.5_{(.1)}$	$55.4_{(.2)}$
None	$37.3_{(.1)}$	$56.1_{(.1)}$	$47.3_{(.6)}$	$53.8_{(.7)}$

Table 3: Results of models with different contextualization techniques on dev sets. We report average scores and standard deviation from five random runs.

3.5 Ablation Studies

To show the contributions of adding graph context and balanced contextualization, we compare results of three different settings (Table 3), i.e., HittER with no context (the context-independent Transformer described in Section 2.1), contextualized HittER without balancing techniques proposed in Section 2.3, and our full model. We find that directly adding in contextual information does not benefit the model ("Unbalanced"), while balanced contextualization generates significantly superior results in terms of MRR on both datasets, especially for the WN18RR dataset, which has a sparser and noisier graph structure.

Breaking down the model's performance by relation types in WN18RR, Table 4 shows that incorporating contextual information brings us substantial improvements on two major relation types, namely the *hypernym* and the *member meronym* relations, which both include many examples belong to the challenging one-to-many relation categories defined in Bordes et al. (2013).

Inferring the relationship between two entities can be viewed as a process of aggregating information from the graph paths between them (Teru et al., 2020). To gain further understanding of what

[8] We do not compare with huge models that employ excessive embeddings size (Lacroix et al., 2018).

Relation Name	Count	No ctx	Full	Gain
hypernym	1174	.144	.181	26%
derivationally related form	1078	.947	.947	0%
member meronym	273	.237	.316	33%
has part	154	.200	.235	18%
instance hypernym	107	.302	.330	9%
synset domain topic of	105	.350	.413	18%
verb group	43	.930	.931	0%
also see	41	.585	.595	2%
member of domain region	34	.201	.259	29%
member of domain usage	22	.373	.441	18%
similar to	3	1	1	0%

Table 4: Dev MRR and relative improvement percentage of our proposed method with or without the *context Transformer* respect to each relation in the WN18RR dataset.

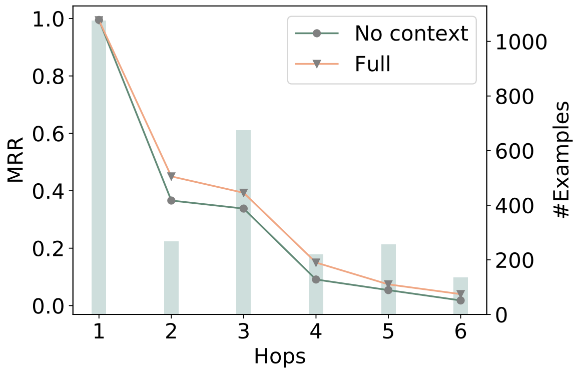

Figure 3: Dev mean reciprocal rank (MRR) in the WN18RR dataset grouped by the number of hops. The bar chart shows the number of examples in each group.

the role the contextual information play from this aspect, we group examples in the development set of WN18RR by the number of hops (*i.e.*, the shortest path length in the undirected training graph) between the subject and the object in each example (Figure 3). From the results, we can see that the MRR metric of each group decreases by the number of hops of the examples. This matches our intuition that aggregating information from longer graph paths is generally harder and such information is more unlikely to be meaningful. Comparing models with and without the contextual information, the contextual model performs much better in groups of multiple hops ranging from two to four. The improvement also shrinks as the number of hops increases.

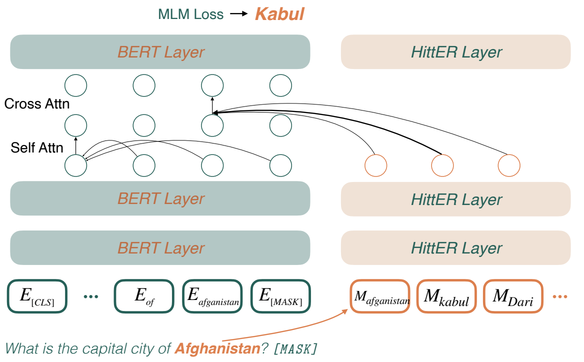
MLM Loss → Kabul
BERT Layer
HittER Layer
Cross Attn
Self Attn
BERT Layer
HittER Layer
BERT Layer
HittER Layer
$E_{[CLS]}$ ··· E_{of} $E_{afganistan}$ $E_{[MASK]}$
$M_{afganistan}$ M_{kabul} M_{Dari} ···
What is the capital city of Afghanistan? [MASK]

Figure 4: Combining HittER and BERT for factoid QA. Each BERT layer is connected to a layer of HittER's context Transformer via a cross-attention module. We jointly fine-tune the combined model to predict the masked entity name in the input question.

4 Factoid QA Experiments

In addition to HittER's superior intrinsic evaluation results, in this section, we conduct a case study on the factoid question answering (QA) task to demonstrate HittER's potential to enhance popular pre-trained Transformer-based language models' performance on knowledge-intensive tasks.

As a Transformer-based model, HittER enables us to integrate its multilayer knowledge representation into other Transformer models (BERT in our case) using the multi-head attention mechanism. In each BERT layer, after the original self-attention module we add a cross-attention module where the queries come from the previous BERT layer while the keys and values come from the output of a corresponding HittER layer (Vaswani et al., 2017), so that HittER's knowledge information can flow into BERT (Figure 4).

We perform experiments on two factoid QA datasets: FreebaseQA (Jiang et al., 2019) and WebQuestionSP (Yih et al., 2016), both pertaining to facts on Freebase. Each question in the two datasets is labeled with a context entity and an inferred relation between the context entity and the answer entity, which we use for preparing the entity and relation inputs for HittER. To better exploit the knowledge in BERT, we follow its pretraining task to create a word-based QA setting, where factoid questions are converted to cloze questions by appending the special `[MASK]` tokens to the end. Both models are trained to recover these `[MASK]` tokens to the original words.[9] We use the BERT-

[9] This is different from the entity-based QA setting. To

	FreebaseQA		WebQuestionSP	
	Full	Filtered	Full	Filtered
Train	20358	3713	3098	850
Test	3996	755	1639	484

Table 5: Number of examples in two Freebase question answering datasets.

Model	FreebaseQA		WebQuestionSP	
	Full	Filtered	Full	Filtered
BERT	$19.8_{(.1)}$	$30.8_{(.1)}$	$23.2_{(.3)}$	$46.5_{(.4)}$
+HittER	$\mathbf{21.2_{(.2)}}$	$\mathbf{37.1_{(.6)}}$	$\mathbf{27.1_{(.2)}}$	$\mathbf{51.0_{(.7)}}$

Table 6: QA accuracy of combining HittER and BERT in two Freebase-based question answering datasets. We report average scores and standard deviation from five random runs.

base model (Devlin et al., 2019) and our best performing HittER model pre-trained on the FB15K-237 dataset. Since FB15K-237 only covers a small portion of Freebase, most questions in the two QA datasets are not related to the knowledge from the FB15K-237 dataset, in which case the input entities for HittER cannot be provided. Thus we also report results under a *filtered* setting, i.e., a subset retaining only examples whose context entity and answer entity both exist on the FB15K-237 dataset.

Our experimental results in Table 6 show that HittER's representation significantly enhances BERT's question answering ability, especially when the questions are related to entities in the knowledge graph used to train HittER. We include more details of the experiments in the Appendix.

5 Related Work

KGE methods have been extensively studied in many diverse directions. Our scope here is limited to methods that purely rely on entities and relations, without access to other external resources.

5.1 Triple-based Methods

Most of the previous work focuses on exploiting explicit geometric properties in the embedding space to capture different relations between entities. Early work uses translational distance-based scoring functions defined on top of entity and relation embeddings (Bordes et al., 2013; Wang et al., 2014; Lin et al., 2015; Ji et al., 2015).

Another line of work uses tensor factorization methods to match entities semantically. Starting from simple bi-linear transformations in the euclidean space (Nickel et al., 2011; Yang et al., 2015), numerous complicated transformations in various spaces have been hence proposed (Trouillon et al., 2016; Ebisu and Ichise, 2018; Sun et al., 2018; Zhang et al., 2019a; Chami et al., 2020; Tang et al., 2020; Chao et al., 2021). Such methods effectively capture the intuition from observation of data but suffer from unobserved geometric properties and are generally limited in expressiveness.

In light of recent advances in deep learning, many more powerful neural network modules such as Convolutional Neural Networks (Dettmers et al., 2018), Capsule Networks (Nguyen et al., 2019), and Transformers (Wang et al., 2019) are also introduced to capture the interaction between entity and relation embeddings. These methods produce rich representations and better performance on predicting missing links in knowledge graphs. However, they are restricted by the amount of information that can be encoded in a single node embedding and the great effort to memorize local connectivity patterns.

5.2 Context-aware Methods

Various forms of graph contexts have been proven effective in recent work on neural networks operating in graphs under the message passing framework (Bruna et al., 2014; Defferrard et al., 2016; Kipf and Welling, 2017). Schlichtkrull et al. (2018, R-GCN) adapt the Graph Convolutional Networks to realistic knowledge graphs which are characterized by their highly multi-relational nature. Teru et al. (2020) incorporate an edge attention mechanism to R-GCN, showing that the relational path between two entities in a knowledge graph contains valuable information about their relations in an inductive learning setting. Vashishth et al. (2020) explore the idea of using existing knowledge graph embedding methods to improve the entity-relation composition in various Graph Convolutional Network-based methods. Bansal et al. (2019) borrow the idea from Graph Attention Networks (Veličković et al., 2018), using a bilinear attention mechanism to selectively gather useful information from neighbor entities. Different from their simple single-layer attention formu-

simplify the modeling architecture, we also make the number of tokens known to all models.

lation, we use the advanced Transformer to capture both the entity-relation and entity-context interactions. Nathani et al. (2019) also propose an attention-based feature embedding to capture multi-hop neighbor information, but unfortunately, their reported results have been proven to be unreliable in a recent re-evaluation (Sun et al., 2020).

6 Conclusion and Future Work

In this work, we proposed HittER, a novel Transformer-based model with effective training strategies for learning knowledge graph embeddings in complex multi-relational graphs. We show that with contextual information from a local neighborhood, our proposed method outperforms all previous approaches in long-standing link prediction tasks, achieving new SOTA results on FB15K-237 and WN18RR. Moreover, we show that the knowledge representation learned by HittER can be effectively utilized by a Transformer-based language model BERT to answer factoid questions.

It is worth mentioning that our proposed balanced contextualization is also applicable to other context-aware KGE methods such as GNN-based approaches. Future work can also apply HittER to other graph representation learning tasks besides link prediction. Currently, our proposed HittER model performs well while only aggregating contextual information from a local graph neighborhood. It would be interesting to extend it with a broader graph context to obtain potential improvements.

Today, the Transformer has become the de facto modeling architecture in natural language processing. As experimental results on factoid question answering tasks showcasing HittER's great potential to be integrated into common Transformer-based model and generate substantial gains in performance, we intend to explore training HittER on large-scale knowledge graphs, so that more NLP models would benefit from HittER in various knowledge-intensive tasks.

Acknowledgments

We thank Hao Cheng, Hoifung Poon, Xuan Zhang, Yu Bai, Aidan San, colleagues from Microsoft Bing Ads team and Microsoft Research, and the anonymous reviewers for their valuable discussions and comments.

References

Ivana Balazevic, Carl Allen, and Timothy Hospedales. 2019. TuckER: Tensor factorization for knowledge graph completion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5185–5194, Hong Kong, China. Association for Computational Linguistics.

Trapit Bansal, Da-Cheng Juan, Sujith Ravi, and Andrew McCallum. 2019. A2N: Attending to neighbors for knowledge graph inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4387–4392, Florence, Italy. Association for Computational Linguistics.

Lisa Bauer, Yicheng Wang, and Mohit Bansal. 2018. Commonsense for generative multi-hop question answering tasks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4220–4230, Brussels, Belgium. Association for Computational Linguistics.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.

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

Samuel Broscheit, Daniel Ruffinelli, Adrian Kochsiek, Patrick Betz, and Rainer Gemulla. 2020. LibKGE - a knowledge graph embedding library for reproducible research. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 165–174, Online. Association for Computational Linguistics.

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2014. Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations*.

Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. 2020. Low-dimensional hyperbolic knowledge graph embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6901–6914, Online. Association for Computational Linguistics.

Linlin Chao, Jianshan He, Taifeng Wang, and Wei Chu. 2021. PairRE: Knowledge graph embeddings via paired relation vectors. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint*

Conference on Natural Language Processing (Volume 1: Long Papers), pages 4360–4369, Online. Association for Computational Linguistics.

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

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Takuma Ebisu and Ryutaro Ichise. 2018. Toruse: Knowledge graph embedding on a lie group. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*.

Hiroaki Hayashi, Zecong Hu, Chenyan Xiong, and Graham Neubig. 2020. Latent relation language models. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 7911–7918.

Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. 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)*, pages 687–696, Beijing, China. Association for Computational Linguistics.

Kelvin Jiang, Dekun Wu, and Hui Jiang. 2019. FreebaseQA: A new factoid QA data set matching trivia-style question-answer pairs with Freebase. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 318–323, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

Timothee Lacroix, Nicolas Usunier, and Guillaume Obozinski. 2018. Canonical tensor decomposition for knowledge base completion. In *International Conference on Machine Learning*, pages 2863–2872.

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

Meng Liu, Hongyang Gao, and Shuiwang Ji. 2020. Towards deeper graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery &amp; Data Mining*, pages 338–348.

Robert Logan, Nelson F. Liu, Matthew E. Peters, Matt Gardner, and Sameer Singh. 2019. Barack's wife hillary: Using knowledge graphs for fact-aware language modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5962–5971, Florence, Italy. Association for Computational Linguistics.

Christopher D Manning, Kevin Clark, John Hewitt, Urvashi Khandelwal, and Omer Levy. 2020. Emergent linguistic structure in artificial neural networks trained by self-supervision. *Proceedings of the National Academy of Sciences*.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Deepak Nathani, Jatin Chauhan, Charu Sharma, and Manohar Kaul. 2019. Learning attention-based embeddings for relation prediction in knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4710–4723, Florence, Italy. Association for Computational Linguistics.

Dai Quoc Nguyen, Thanh Vu, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung. 2019. A capsule network-based embedding model for knowledge graph completion and search personalization. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2180–2189, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 43–54, Hong Kong, China. Association for Computational Linguistics.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 74–84, Atlanta, Georgia. Association for Computational Linguistics.

Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. 2020. Dropedge: Towards deep graph convolutional networks on node classification. In *International Conference on Learning Representations*.

Daniel Ruffinelli, Samuel Broscheit, and Rainer Gemulla. 2020. You can teach an old dog new tricks! on training knowledge graph embeddings. In *International Conference on Learning Representations*.

Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. 2020. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4498–4507, Online. Association for Computational Linguistics.

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

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2018. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*.

Zhiqing Sun, Shikhar Vashishth, Soumya Sanyal, Partha Talukdar, and Yiming Yang. 2020. A re-evaluation of knowledge graph completion methods. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5516–5522, Online. Association for Computational Linguistics.

Yun Tang, Jing Huang, Guangtao Wang, Xiaodong He, and Bowen Zhou. 2020. Orthogonal relation transforms with graph context modeling for knowledge graph embedding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2713–2722, Online. Association for Computational Linguistics.

Komal K Teru, Etienne Denis, and William L Hamilton. 2020. Inductive relation prediction by subgraph reasoning. In *Proceedings of the 37th International Conference on Machine Learning*.

Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 57–66, Beijing, China. Association for Computational Linguistics.

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

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. 2020. Composition-based multi-relational graph convolutional networks. In *International Conference on Learning Representations*.

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

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Pat Verga, Haitian Sun, Livio Baldini Soares, and William Cohen. 2021. Adaptable and interpretable neural MemoryOver symbolic knowledge. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3678–3691, Online. Association for Computational Linguistics.

Haoyu Wang, Vivek Kulkarni, and William Yang Wang. 2020. Dolores: Deep contextualized knowledge graph embeddings. In *Automated Knowledge Base Construction*.

Hongwei Wang, Hongyu Ren, and Jure Leskovec. 2021. Relational message passing for knowledge graph completion. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery &amp; Data Mining*, KDD '21, page 1697–1707, New York, NY, USA. Association for Computing Machinery.

Quan Wang, Pingping Huang, Haifeng Wang, Songtai Dai, Wenbin Jiang, Jing Liu, Yajuan Lyu, Yong Zhu, and Hua Wu. 2019. Coke: Contextualized knowledge graph embedding. *arXiv preprint arXiv:1911.02168*.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 1112–1119.

Robert West, Evgeniy Gabrilovich, Kevin Murphy, Shaohua Sun, Rahul Gupta, and Dekang Lin. 2014. Knowledge base completion via search-based question answering. In *Proceedings of the 23rd international conference on World wide web*, pages 515–526.

Wenhan Xiong, Thien Hoang, and William Yang Wang. 2017. DeepPath: A reinforcement learning method for knowledge graph reasoning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 564–573, Copenhagen, Denmark. Association for Computational Linguistics.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *International Conference on Learning Representations*.

Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–206, Berlin, Germany. Association for Computational Linguistics.

Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. 2019a. Quaternion knowledge graph embeddings. In *Advances in Neural Information Processing Systems*, pages 2735–2745.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019b. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy. Association for Computational Linguistics.

A Embedding Clustering

Table 7 lists the entity clustering results of first few entities in each dataset, based on our learned entity representations. Clusters in FB15K-237 usually are entities of the same type, such as South/Central American countries, government systems, and American voice actresses. While clusters in WN18RR are generally looser but still relevant to the topic of the central word.

B Factoid QA Experiment Details

In order to connect our HittER model with BERT, we add a cross-attention module after the self-attention module in each BERT layer. Following the encoder-decoder attention mechanism in Vaswani et al. (2017), we use queries from previous BERT layer and keys and values from the output of a corresponding HittER layer. The pre-trained BERT (base) and HittER models we use have two differences in terms of hyper-parameter settings, i.e., the number of layers and dimentionality. Since BERT has 12 layers while HittER only has 6 layers, we connect every two BERT layers to one HittER layer and skip the first two layers in BERT.[10] Before attention computation, we increase the dimentionality of HittER's output representations to the number of BERT's using linear transformations. The dimensionality and number of cross-attention heads are set as the same configuration of the BERT base model we use.

We finetune all of our question answering (QA) models using a batch size of 16 for 20 epochs. We use the Adam optimizer (Kingma and Ba, 2015) with a learning rate of $5e-6$ for all pretrained weights and a learning rate of $5e-5$ for newly added cross-attention modules. The learning rate linearly increases from 0 over the first 10% training steps.

C Discussion

C.1 Right Context for Link Prediction

Structural information of knowledge graphs can come from multiple forms, such as graph paths, sub-graphs, and the local neighborhood that we used in this work. In addition, these context forms can be represented in terms of the relation type, the entity, or both of them.

[10] Among various connection strategies, this strategy gives us the best results in pilot experiments, which also suggests that HittER stores different types of information in its multi-layer representations.

Entity	Top 5 Neighbors
Dominican Republic	Costa Rica, Ecuador, Puerto Rico, Colombia, El Salvador
Republic	Presidential system, Unitary state, Democracy, Parliamentary system, Constitutional monarchy
MMPR	Power Rangers, Sonic X, Ben 10, Star Trek: Enterprise, Code Geass
Wendee Lee	Liam O'Brien, Michelle Ruff, Hilary Haag, Chris Patton, Kari Wahlgren
Drama	Thriller, Romance Film, Mystery, Adventure Film, LGBT
Land reform	Pronunciamento, Premium, Protest march, Reform, Birth-control reformer
Reform	Reform, Land reform, Optimization, Self-reformation, Enrichment
Cover	Surface, Spread over, Bind, Supply, Strengthen
Covering	Sheet, Consumer goods, Flap, Floor covering, Coating
Phytology	Paleobiology, Zoology, Kingdom fungi, Plant life, Paleozoology

Table 7: Nearest neighbors of first five entities in FB15K-237 and WN18RR based on the cosine similarity between learned entity embeddings from our proposed method.

In this work, we show that a simple local neighborhood is sufficient to greatly improve a link prediction model. In early experiments in the FB15K-237 dataset, we actually observe that masking out the source entity all the time does not harm the model performance much. This implies that the contextual information in a dense knowledge graph like FB15K-237 is rich enough to replace the source entity in the link prediction task.

Recently, Wang et al. (2021) argue that graph paths and local neighborhood should be jointly considered when only the relation types is used (throwing out entities). Although some recent work has made a first step towards utilizing graph paths for knowledge graph embeddings (Wang et al., 2019, 2020), there is still no clear evidence of its effectiveness.

C.2 Limitations of the *1vsAll* Scoring

Recall that HittER learns a representation for an incomplete triplet (e_s, r_p) and then computes the dot-product between it and all the candidate target entity embeddings. This two-way scoring paradigm, which is often termed *1vsAll* scoring, supports fast training and inference when the interactions between the source entity and the predicate are captured by some computation-intensive operations (*i.e.*, the computations of Transformers in our case), but unfortunately loses three-way interactions. We intentionally choose 1vsAll scoring for two reasons. On one hand, 1vsAll together with cross-entropy training has shown a consistent improvement over other alternative training configurations empirically (Ruffinelli et al., 2020). On the other hand, it ensures a reasonable speed for the inference stage where the 1vsAll scoring is necessary.

Admittedly, early interactions between the source entity and the target entity can provide valuable information to inform the representation learning of the incomplete triplet (e_s, r_p). For instance, we find that a simple bilinear formulation of the source entity embeddings and the target entity embeddings can be trained to reflect the distance (measured by the number of hops) between the source entity and the target entity in the graph. We leave the question of how to effectively and efficiently incorporate such early fusion for future work.