Neural Processing Letters
https://doi.org/10.1007/s11063-021-10586-8

Siamese Pre-Trained Transformer Encoder for Knowledge Base Completion

Mengyao Li[1] · Bo Wang[2] · Jing Jiang[1]

Accepted: 8 July 2021

Abstract

In this paper, we aim at leveraging a Siamese textual encoder to efficiently and effectively tackle knowledge base completion problem. Traditional graph embedding-based methods straightforwardly learn the embeddings by considering a knowledge base's structure but are inherently vulnerable to the graph's sparsity or incompleteness issue. In contrast, previous textual encoding-based methods capture such structured knowledge from a semantic perspective and employ deep neural textual encoder to model graph triples in semantic space, but they fail to trade off the contextual features with model's efficiency. Therefore, in this paper we propose a Siamese textual encoder operating on each graph triple from the knowledge base, where the contextual features between a head/tail entity and a relation are well-captured to highlight relation-aware entity embedding while a Siamese structure is also adapted to avoid combinatorial explosion during inference. In the experiments, the proposed method reaches state-of-the-art or comparable performance on several link prediction datasets. Further analyses demonstrate that the proposed method is much more efficient than its baseline with similar evaluating results.

Keywords Knowledge base completion · Pre-trained transformer encoder · Siamese network

✉ Bo Wang
bowang19@mails.jlu.edu.cn

Mengyao Li
Mengyao.Li-1@student.uts.edu.au

Jing Jiang
Jing.Jiang@uts.edu.au

[1] Australian AI Institute, School of Computer Science, FEIT, University of Technology Sydney, Ultimo, Australia

[2] School of Artificial Intelligence, Jilin University, Jilin, China

Published online: 06 September 2021

1 Introduction

Knowledge base completion (KBC) over knowledge base[1] [1] is crucial in natural language processing (NLP) literature, which including link prediction, relation classification and triple classification. It aims at automatically complete a graph triple, i.e., (head entity, relation, tail entity), from the curated knowledge base that suffers from sparsity or incompleteness issue, and thus benefits to a wide range of NLP tasks depending heavily on knowledge base, such as knowledge-based question answering [22], information retrieval [11], recommendation system [33], etc.

Recent works to tackle knowledge base completion tasks can be coarsely categorized into two groups. The first one is *graph embedding approach*, e.g., RotatE [24] and ConvE [5], to assign each entity and relation in knowledge base with a real-valued, low-dimensional representation vector. These works optimize the vectors by considering their geometric relationship inside a graph triple (e.g., distance metrics) in an embedding space. However, these works only explore structured or relational information in observed triples, and thus inherently suffer from knowledge base's sparsity or incompleteness problem.

On the other hand, the second group refers to *textual encoding approach* [29] that integrates textual information of the entities and relations (e.g., their node contents and textual mentions) with the structured knowledge. This paradigm can be very effective when an expressively powerful text encoder, especially the pre-trained [6,10] Transformer encoder [27], is applied to the texts of a graph triple.

Nevertheless, how to trade off the efficiency with contextualization is still an open question: (1) Individually considering the textual information of entities or relations is very efficient but suffers from a lack of contextualized features. An intuition behind this that, an entity can express different meanings in different domains and the domain information can be easily implied from the relation in the same graph triple. For example, give a relation named "*BirthPlace*", its head entity is constrained to "*Person*" and its tail entity is constrained to "*Place*". (2) At the other extreme, recent works are prone to straightforwardly apply a pre-trained language model to the texts of a graph triple, where texts of head entity, relation and tail entity are directly concatenated with special token to separate. These methods, however, require enormous computational overheads due to combinatorial explosion during inference of link prediction. This is caused by, given a head (tail) entity and a certain relation, a model needs to distinguish the most plausible tail (head) entity among all entities in a graph, and the number of distinct entities in a graph can be millions.

In this paper, we aim to achieve a balanced trade-off between the efficiency during inference and the effectiveness with contextualized features. To this end, we propose a novel Siamese network for knowledge base completion. Inspired by Siamese network [3,34] and Sentence- [19], a graph triple is split into two parts and a Siamese encoder is applied to each part to generate the corresponding representation. And a neural classifier based on the two representations derived from the two parts is used to predict the plausibility of the input graph triple. The proposed Siamese architecture significantly improve the efficiency of link prediction's inference that can be formulated as an information retrieval procedure [19]. And the contextual information across head/tail entity and relation is also taken into account to provide rich domain knowledge, which naturally mitigates the ambiguity issue in knowledge base.

[1] In this paper, "knowledge base" and "knowledge graph" are interchangeable, denoting curated graphs such as Freebase and WordNet.

In the experiments, we evaluate our proposed approach on several link prediction benchmarks and show new state-of-the-art performance. The improvement in terms of Mean Rank (MR) is especially remarkable, and further analysis shows our model can deliver consistent performance, even with the entity never presenting in training phase. Furthermore, compared to the method directly applying an encoder to a concatenation of a triple's texts, our approach can significantly reduce the inference time.

The main contributions of this work can be summarized as

– We propose to leverage the pre-trained encoder for handling knowledge base completion from a semantic perspective, with considering rich contextualized information of entities and relations.
– We propose to balance between efficiency of inference and effectiveness of contextualization, by employing a Siamese architecture with subtly designed triple inputs.
– We evaluate the proposed method extensively on several benchmark datasets of link prediction, which achieves state-of-the-art or competitive performance.

2 Prerequisite

This section begins with a brief introduction to the recently-promoted pre-trained language model. Then, the baseline of our proposed model, KG-BERT, is introduced in summary.

2.1 Pre-Trained Transformer Encoder

Training a textual encoder in a self-supervised manner has been proved effective to initialize the model, and thus benefits a wide range of natural language processing tasks by fine-tuning. As illustrated in Fig. 1, Bidirectional Encoder Representations from Transformers (BERT) model is first pre-trained by two self-supervised objectives, i.e., masked language modeling (MLM) and next sentence prediction (NSP), where the training samples are merely from raw corpora without any human-annotated labels. After being pre-trained, the Transformer encoder [27] can be adapted to tackle most of natural language processing tasks, such as multi-genre natural language inference (NLI), named entity recognition (NER) and machine reading comprehension (MRC). The two self-supervised learning tasks to pre-train a BERT will be formally introduced in the following.

Give a piece of text (e.g., natural language sentence), s, a tokenizing algorithm is invoked to obtain a sequence of tokens, i.e., $s = [x_1, x_2, \ldots, x_N]$. Based on this, the first self-supervised objective, masked language modeling, randomly masks 15% of the total tokens: of those, 80% replaced with a special token `[MASK]`, 10% replaced with a random token uniformly sampled from vocabulary and 10% kept unchanged. We denote the masked sequence as $\hat{s} = [\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_N]$. Then, the masked sequence is passed into a Transformer encoder [27] to produce contextualized representations for each (masked) token, which can be written as

$$\boldsymbol{C} = \text{Trans} - \text{Enc}([\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_N]) \in \mathbb{R}^{d \times N}, \tag{1}$$

where $\boldsymbol{C} = [\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_N]$ is the sequence of resulting contextualized representations. Lastly, the training loss of MLM is

$$\mathcal{L}^{(MLM)} = -\sum_{i \in \mathcal{M}} \log P(x_i | \boldsymbol{c}_i) := \text{softmax}(\text{MLP}(\boldsymbol{c}_i; \theta^{(1)})), \tag{2}$$

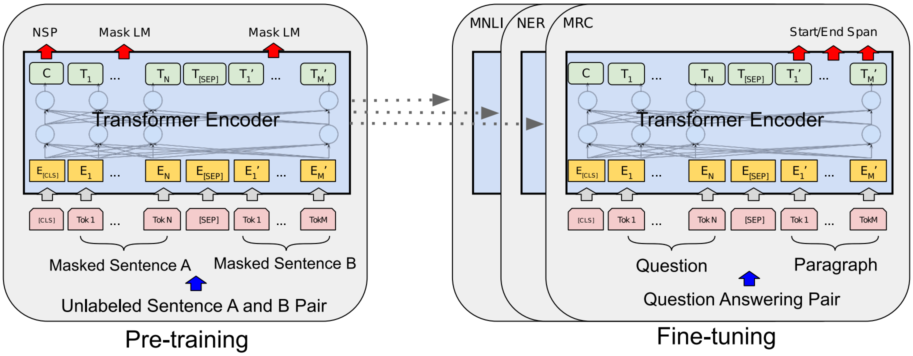

Fig. 1 Bidirectional Encoder Representations from Transformers (BERT) pre-trained on two self-supervised tasks (left) and then fine-tuned on a variety of natural language processing tasks (right)

where $\mathcal{M}$ denotes the masked indices for the sequence. Compared to the casual language model auto-regressively predicting the next token, the masked language model considers both sides of the contexts to predict the masked token. On the other hand, being jointly trained with the MLM objective, the second one, next sentence generation, aims at discriminating whether the two input sentences are in coherent context (i.e., adjacent). Formally, considering two masked sentences, $\hat{s}^{(1)} = [\hat{x}_1^{(1)}, \hat{x}_2^{(1)}, \ldots, \hat{x}_N^{(1)}]$ and $\hat{s}^{(2)} = [\hat{x}_1^{(2)}, \hat{x}_2^{(2)}, \ldots, \hat{x}_N^{(2)}]$, they are concatenated with pre-defined special tokens and then passed into the Transformer encoder, i.e.,

$$\boldsymbol{v} = \text{Pool}(\text{Trans} - \text{Enc}([\texttt{[CLS]}, \hat{x}_1^{(1)}, \ldots, \hat{x}_N^{(1)}, \texttt{[SEP]}, \hat{x}_1^{(2)}, \ldots, \hat{x}_N^{(2)}, \texttt{[SEP]}])), \quad (3)$$

where `[CLS]` and `[SEP]` are special tokens to indicate "start of input" and "separator" respectively, as defined by [6], and $\text{Pool}(\cdot)$ denotes collecting the contextualized representation of `[CLS]` to represent the whole input sequence, which can be used to fulfill task-specific objective. On the NSP task, the representation vector is fed into a binary classifier to judge whether the two-sentence is adjacent, i.e.,

$$\boldsymbol{p}^{(NSP)} = \text{softmax}(\text{MLP}(\boldsymbol{v}; \theta^{(2)})) \in \mathbb{R}^2. \quad (4)$$

And the loss function to optimize is defined as

$$\mathcal{L}^{(NSP)} = -(y \log \boldsymbol{p}^{(NSP)}_{(\hat{y}=1)} + (1 - y)\boldsymbol{p}^{(NSP)}_{(\hat{y}=0)}). \quad (5)$$

Lastly, $\mathcal{L}^{(MLM)}$ and $\mathcal{L}^{(NSP)}$ are jointly optimized over large-scale raw corpora to learn an expressively powerful Transformer encoder. The encoder is able to produce generic contextual representations and thus can be fine-tuned on various downstream natural language processing tasks.

2.2 KG-BERT

In light of the pre-trained Transformer encoder, KG-BERT [32] straightforwardly applies the encoder to a graph triple for knowledge base completion. In particular, the textual information of a graph triple's components, i.e., head entity, relation and tail entity, is incorporated by following [6]. An illustration of KG-BERT is shown in Fig. 2. Formally, considering $s^{(h)} = [x_1^{(h)}, \ldots, x_N^{(h)}]$, $s^{(r)} = [x_1^{(r)}, \ldots, x_N^{(r)}]$, $s^{(t)} = [x_1^{(t)}, \ldots, x_N^{(t)}]$ as the texts for the

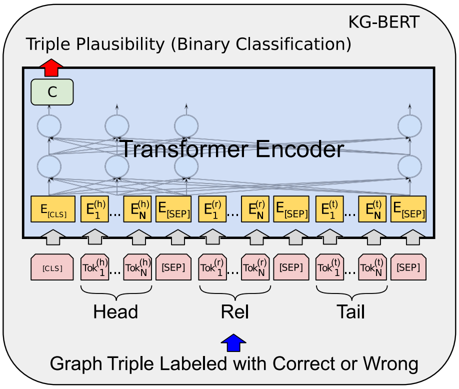

Fig. 2 KG-BERT model that applies pre-trained Transformer encoder to knowledge base completion tasks

components, a concatenation of them with special token as separator is passed into the Transformer encoder, i.e.,

$$s^{(tp)} = [\texttt{[CLS]}, x_1^{(h)}, \ldots, x_N^{(h)}, \quad \texttt{[SEP]}, x_1^{(r)}, \ldots, x_N^{(r)}, \texttt{[SEP]}, x_1^{(t)}, \ldots, x_N^{(t)}, \texttt{[SEP]}],$$
$$\boldsymbol{v}^{(tp)} = \text{Pool}(\text{Trans} - \text{Enc}(s^{(tp)})). \tag{6}$$

Then, the contextualized representation $\boldsymbol{v}^{(tp)}$ of the whole triple is passed into a multi-layer perceptron (MLP) to discriminate whether the input triple is correct or not. This general pipeline can solve all tasks in knowledge base completion, including triple classification, triple classification and link prediction, whereas only differs in data pre-processing [32]. However, when handling link predictions, KG-BERT needs huge computational overheads during inference due to combinatorial explosion.

3 Siamese Encoder for Knowledge Base Completion

Inspired by sentence-BERT [19] using Siamese architecture for efficient sentence matching, we propose to employ Siamese textual encoder to accelerate the inference of link prediction. The architecture of the proposed model is shown in Fig. 3.

3.1 Siamese Encoder

As shown in KG-BERT [32], rich contextual features between head/tail entity and relation are crucial to derive domain- or range-specific contextualized representations for the entities. To a certain degree, it is able to alleviate the ambiguity issue in knowledge base. To preserve the contextual features in a Siamese encoder, we propose to split a graph triple into two parts: the first part consists of the head entity and relation, and the second part consists of the relation and tail entity. Although the relation in a graph triple will be encoded twice by two branches of the Siamese encoder, the computational cost would not significantly change compared to KG-BERT baseline. This is because the computational complexity of the Transformer encoder is quadratically proportional to the length of the input sequence. Formally, the two parts of a graph triple are encoded via

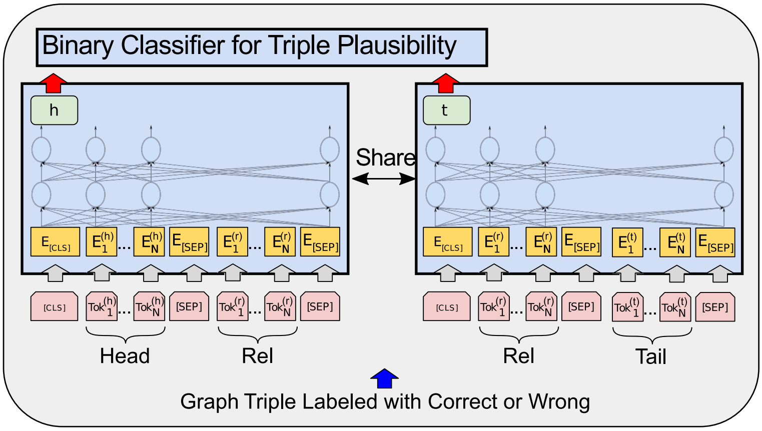

Fig. 3 The proposed Siamese encoder for knowledge base completion. "Share" denotes the parameters in the two branches of encoders are shared

$$\boldsymbol{v}^{(h)} = \text{Pool}(\text{Trans} - \text{Enc}([\texttt{[CLS]}, x_1^{(h)}, \ldots, x_N^{(h)}, \quad \texttt{[SEP]}, x_1^{(r)}, \ldots, x_N^{(r)}, \texttt{[SEP]}]),$$
$$\boldsymbol{v}^{(t)} = \text{Pool}(\text{Trans} - \text{Enc}([\texttt{[CLS]}, x_1^{(r)}, \ldots, x_N^{(r)}, \quad \texttt{[SEP]}, x_1^{(t)}, \ldots, x_N^{(t)}, \texttt{[SEP]}]).$$

To indicate whether a sequence from an entity or a relation, we employ the segment identifier [6] and set 0 for the tokens from an entity and 1 for those from a relation. Hence, the identifier for the first part is $[0, \ldots, 0, 1, \ldots, 1]$, while that for the second part is $[1, \ldots, 1, 0, \ldots, 0]$.

3.2 Learning Objective

Based on the two contextualized representations from Siamese encoder, there are several options to integrate them for a classification task [19]. Here, we adopt a naive way to integrate them. That is, we directly concatenate the two representations, and pass the concatenation into an MLP to fulfill the binary classification task, i.e.,

$$\boldsymbol{p}^{(tp)} = \text{softmax}(\text{MLP}([\boldsymbol{v}^{(h)}; \boldsymbol{v}^{(t)}], \theta)) \in \mathbb{R}^2. \tag{7}$$

Before formulating the loss function, a negative sampling must be performed to obtain the wrong graph triples corresponding to the correct one [24]. We leverage a vanilla negative scheme that randomly selects an entity from the knowledge base to corrupt the head or tail entity in a graph triple, as long as the corrupted triple does not exist in the knowledge base. The loss function of the proposed method is formulated as

$$\mathcal{L} = -\frac{1}{|\mathcal{D}|} \sum_{tp+\in\mathcal{D}} \left(\alpha \log \boldsymbol{p}_{(y=1)}^{(tp+)} + \log \boldsymbol{p}_{(y=0)}^{(tp-)}\right), \tag{8}$$

where, "$tp-$" is a wrong triple corrupted from "$tp+$" which is in real world dataset $\mathcal{D}$, and $\boldsymbol{p}^{(tp+)}$ and $\boldsymbol{p}^{(tp-)}$ are calculated via Eq (7) for correct and wrong triples respectively. Note, α is a hyperparameter to balance the weights of positive and negative examples. Empirically, we set α to 0.2 for link prediction task, making the model incline to negative example, since during inference, the model needs to distinguish the only positive example among thousands of candidates.

3.3 Model Efficiency

When performing link prediction over a large-scale knowledge base, our proposed model based on the Siamese network can be much more efficient than the non-Siamese model, e.g., KG-BERT. To quantify the acceleration, we coarsely conduct an analysis in the following. First of all, it is worth mentioning that the major overheads of Transformer-based encoder come from the encoder itself and grow quadratically with the input sequence.

We use E and R to denote the numbers of unique entities and relations in the knowledge base respectively, and use S to stand for the length of the text of each component in a graph triple. In training phrase, due to the length of a relation's text is much shorter than an entity's in general, Siamese encoder can achieve 2 times speedup for one triple than KG-BERT, i.e., $O(S^2/2) \leftarrow O(2(S/2)^2)$ vs. $O(S^2)$ for one triple. In inference stage, to complete the inference over the whole graph, KG-BERT requires $O(9S^2E^2R) \leftarrow O((3S)^2E^2R)$ whereas the Siamese encoder only requires $O(8S^2ER) \leftarrow O(2(2S)^2ER)$. Hence, our proposed theoretically improves the inference efficiency by about E times for link prediction, where the E is up to millions.

4 Experiments

In this section, we first evaluate the proposed Siamese encoder on popular link prediction benchmarks for knowledge base completion. Then, we conduct a series of analyses for insights into the proposed model.

4.1 Experimental Setups

Datasets. Two link prediction benchmark datasets are used to evaluate our approach. The first one is WN18RR [5], a subgraph of WordNet [13], which includes natural language words/phrases as vertices and their semantic relations (e.g., hypernym and synonym) as edges. The second one is FB15k-237 [25], a subgraph of Freebase, which includes named entities as vertices and their factoid relations as edges. The statistics of these two datasets are summarized in Table 1. As for transforming ontology in KB to text, we employ textual description of the entities and textual content of the relations as textual input to the Siamese encoder, by following [32]. It is noteworthy that, as first stated by [25] and confirmed by [5], WN18 and FB15k suffer from informative value, which causes $>80\%$ of the test triples (e^1, r^1, e^2) can be found in the training set with another relation: (e^1, r^2, e^2) or (e^2, r^2, e^1). [5] used a rule-based model which learned the inverse relation and achieved state-of-the-art results on the dataset. It should therefore not used for research evaluation anymore. Therefore, WN18RR and FB15k-237, derived from WN18 [1] and FB15k [1] respectively by removing inverse relations and data leakage, are the most popular and representative benchmark datasets for link prediction [5,32].
Training Setups. We use BERT or RoBERTa to initialize the Transformer encoder. And we optimize the proposed model via mini-batch SGD and employ Adam algorithm as the optimizer. As for hyperparameters, we set the batch size to 64, the number of training epochs to 7; and we set learning rate to 5×10^{-5} for BERT-initialized model and 1×10^{-5} for RoBERTa-initialized model, where learning rate warmup strategy is also leveraged with a period of 5%. Note, we do not validate our model on dev set during training since the

Table 1 Statistics of WN18NN and FB15k-237 datasets for link prediction

Dataset	Ent Num	Rel Num	Train Num	Dev Num	Test Num
WN18RR	40,943	11	86, 835	3034	3134
FB15k-237	14,541	237	272, 115	17, 535	20, 466

validation is time-consuming, and only use the last checkpoint to derive evaluating metrics on test set.

Evaluating Metrics. The setting of link prediction is that, given an oracle triple from dev/test set, we first use all other entities to corrupt its head or tail entity, and rank the only oracle triple against all other corrupted triples. Note that we use the "filter" setting [1] to remove all positive triples except current oracle test triple. Since the essence of link prediction is a ranking problem, there are five metrics regarding the rank of the only positive triple, which can be grouped into two classes. The first class includes mean rank (MR) and mean reciprocal rank (MRR) whereas the second class includes Hits@N (commonly @1 @3 & @10) denoting the proportion of oracle test triple ranked into top-N.

Baseline and Competitors. We list the proposed method's baseline and its competitive approaches in the following and refer readers to the original papers for more details and more comparable approaches.

- **SiaEnc** is the proposed Siamese Encoder in this paper for knowledge base completion.
- **KG-BERT** [32] is the baseline of the proposed approach, which takes a concatenation of the head entity, relation and tail entity as input to the Transformer encoder.
- **TransE** [1] is a graph embedding approach, aiming at learning the embedding of entities and relations by considering the "translation" relationship in a triple. In TransE, they use $-||\boldsymbol{h}+\boldsymbol{r}-\boldsymbol{t}||$ as score function and maximize the score margin of positive triple again its corrupted one.
- **RotatE** [24], as a state-of-the-art upgrade of TransE, uses complex vectors to represent entities and vector rotate to define the relation. They use $-||\boldsymbol{h}\circ\boldsymbol{r}-\boldsymbol{t}||$ as score function, which can model various relations in KB, including symmetry, antisymmetry, inversion and composition.
- **ConvE** [5], instead of using a translation formula as in TransE and RotatE, employs convolutional neural networks (CNNs) to model their relations. Its score function can be briefly written as $\text{CNN}([\boldsymbol{h};\boldsymbol{r}])^T\boldsymbol{t}$.
- **R-GCN** [20] employs graph convolutional neural network with a consideration of relation to enrich the representations of the graph, which can be followed by any translation-based method (e.g., DistMult [31] in the paper).
- **Type Constraints** [9], **TypeCons** for short, is a new setting of link prediction, which limits the head and tail corruption candidates to relation-specific domain and range respectively. So it can be integrated into any link prediction methods to improve the performance.

4.2 Main Evaluations

As shown in Table 2, the proposed Siamese Encoder for knowledge base completion is able to deliver state-of-the-art or competitive performance over most of evaluating metrics on WN18RR and FB15k-237. In particular, compared to traditional graph embedding approaches including TransE, ConvE, RotatE, R-GCN, etc., our approach can achieve decent performance, and some improvements are observed. The improvement is especially remarkable in terms of mean rank (MR), and the reason behind this will be analyzed in the next section. In contrast to our approach's baseline KG-BERT, our approach delivers a complete improvement, and sets new state-of-the-art results in the textual encoding genre.

Furthermore, to comprehensively compare the proposed SiaEnc and KG-BERT, we show the link prediction evaluation results with different model initializations (i.e., BERT-base and RoBERTa-large). As shown in Table 3, although the proposed SiaEnc performs slightly worse than KG-BERT when using BERT-base as the initialization of the Transformer encoder, it overall surpasses KG-BERT with RoBERTa-large except slightly underperform in terms of Hits@10.

Our proposed SiaEnc only reaches similar performance with KG-BERT, but as shown in Table 4 SiaEnc is much more efficient during both training and inference. Specifically, even though the relation's tokens are encoded twice in the proposed SiaEnc, the training time (per epoch) is even less than the vanilla approach. We also empirically improve the inference efficiency by $1 \sim 2$ magnitudes: $18\times$ for base model and $43\times$ for large model.

4.3 Improvement Analysis

To figure out the reason why the proposed SiaEnc brings huge promotions in terms of mean rank (MR), we conduct a quantitative analysis by comparing metrics on the original test set with those on a new setting. In the new setting ("Unseen Test" in Table 5), all the test triples involving the entities that do not exist in the training phase are kept, otherwise removed. A straightforward intuition behind such settings is that given an entity never appears in the training set, its embedding in the traditional graph embedding approach (e.g., ConvE and RotatE) is savage at random. By comparison, SiaEnc starts with tokens composing entities and relations. What's more, the sub-word tokenizing algorithm [6,10] further mitigates the out-of-vocabulary (OOV) problem, making the model more robust to the entities never showing during training. Hence the proposed method alleviates the problem, which is reflected in the substantial improvement in mean rank (MR) since MR is an un-normalized metric varying from one to thousands.

As shown in Table 5, in addition to our approach, we also evaluate RotatE [24] in both settings, which is a recently-proposed graph embedding approach for link prediction. It is observed that RotatE degenerates to a random model, showing significant performance drop in all the metrics (close to zero). In contrast, the proposed SiaEnc surprisingly achieves even better effectiveness in Hits@1, Hits@3 and MRR. It is worth noticing that, $\sim 7\%$ (209/3134) and $\sim 0.1\%$ (29/20466) of test triples contain the unseen entities on WN18RR and FB15k-237 respectively. This explains why the improvement is significant on the former whereas marginal on the later, compared to traditional graph embedding approaches.

4.4 Ablation Study

We also conduct an ablation study in Table 6 to verify the effectiveness of each module in the proposed model. In particular, we first change the binary classification objective into a ranking one with hinge loss as in traditional graph embedding approach, which shows 0.172 drop on Hits@10. Then, we imitate the metric in translation-based graph embedding and formulate the score function as $||\boldsymbol{v}^{(h)} - \boldsymbol{v}^{(t)}||_2$, which also leads to 0.082 decrease on Hits@10. Next, we cut off the parameter sharing between the two branches of encoders, and witness 0.07 of Hits@10 drop. Although the model without sharing achieves a better Hits@1

Table 2 Link prediction performance on WN18RR and FB15k-237 benchmark datasets. Our proposed approach, SiaEnc, is initialized by RoBERTa-large

	WN18RR					FB15k-237				
	Hits@1	Hits@3	Hits@10	MR	MRR	Hits@1	Hits@3	Hits@10	MR	MRR
R-GCN [20]†	.080	.137	.207	6700	.123	.100	.181	.300	600	.164
SACN [21]	.430	.480	.540	–	.470	.270	.400	.550	–	.360
ConvE [5]†	.419	.470	.531	4464	.456	.225	.341	.497	245	.312
ConvKB [15]	.058	.445	.558	1295	.265	.198	.324	.471	216	.289
CapsE [16]	–	–	.560	719	.415	-	-	.593	303	.523
TransE [1]†	.042	.441	.532	2300	.243	.198	.376	.441	323	.279
DistMult [31]†	.412	.470	.504	7000	.444	.199	.301	.446	512	.281
ComplEx [26]†	.409	.469	.530	7882	.449	.194	.297	.450	546	.278
RotatE [24]	.428	.492	.571	3340	.476	.241	.375	.533	177	.338
QuatE [35]	.436	.500	.564	3472	.481	.221	.342	.495	176	.311
QuatE + TypeCons	.438	.508	.582	2314	.488	.248	.382	.550	87	.348
KG-BERT [32]	.041	.302	.524	97	.216	-	-	.420	153	-
SiaEnc (ours)	.143	.437	.689	62	.329	.170	.299	.465	121	.268
+ TypeCons	.188	.507	.744	33	.383	.242	.369	.538	73	.339

† denotes the metric values are copied from [14] after re-evaluation. Others are taken from their original papers

Table 3 Comparisons with KG-BERT baseline on WN18RR with different initializations for the Transformer encoder

	Hits@1	Hits@3	Hits@10	MR	MRR
KG-BERT (BERT-base)	.041	.302	.524	97	.216
SiaEnc (BERT-base)	.160	.271	.471	514	.254
KG-BERT (RoBERTa-large)	.119	.387	.698	95	.297
SiaEnc (RoBERTa-large)	.143	.437	.689	62	.329

Table 4 Empirical running times of both training and inference on WN18RR, with different initializations

	Time/Epoch (m)	Inference time (h)
KG-BERT (BERT-base)	40	32
SiaEnc (BERT-base)	32	1.7
KG-BERT (RoBERTa-large)	79	92
SiaEnc (RoBERTa-large)	75	2.1

The time information is collected when running the implementation on a single NVIDIA RTX 6000 GPU with mixed float computation precision

Table 5 Improvement analysis according to evaluations on WN18RR

Method	Setting	Hits@1	Hits@3	Hits@10	MR	MRR
RotatE [24]	Normal Test	.428	.492	.571	3340	.476
	Unseen Test	.005	.007	.012	17,955	.007
	Δ	**−0.42**	**−0.49**	**−0.56**	**−14,615**	**−0.47**
SiaEnc	Normal Test	.143	.437	.689	62	.329
	Unseen Test	.264	.474	.690	175	.405
	Δ	*+0.12*	*+0.04*	∼	**−113**	*+0.08*

"Unseen Test" refers to keeping the test triples involving the entities that do not exist in the training phase
Bold and italic respectively denote the degradation and improvement of the "Unseen Test" setting on all metrics compared with the "Normal Test" setting

Table 6 Ablation study on WN18RR

	Hits@1	Hits@3	Hits@10	MR	MRR
SiaEnc (RoBERTa-large)	.143	.437	.689	62	.329
· replacing *cls* w/ *rank*	.155	.355	.517	217	.289
· replacing *MLP* w/ *L2*	.105	.398	.607	122	.281
· w/o parameter-sharing	.214	.419	.619	221	.348
· w/o Siamese	.041	.302	.524	97	.216

result, it requires approximately 2× learnable parameters and thus significantly increases the space complexity. Lastly, we remove the Siamese structure, and the model degenerates as the KG-BERT model, which exhibits significant performance drop.

4.5 Error Analysis and Case Study

Entity Ambiguity. The problem of entity ambiguity is ubiquitous in most of the curated knowledge bases. It means an entity could express totally different meanings in different contexts. This problem hurts the performance of a textual encoder; especially the given contexts of an entity or relation are always incomplete. For example, an entity/word "*rush*" has the following *seven* meanings in WN18RR:

- *attack suddenly*;
- *cause to move fast or to rush or race*;
- *a sudden burst of activity*;
- *act or move at high speed*;
- *urge to an unnatural speed*;
- *grass-like plants growing in wet places and having cylindrical often hollow stems*;
- *physician and American Revolutionary leader*.

Note they are regarded as different graph nodes/entities in the WN18RR. It is observed that some meanings of them are very similar, which essentially challenge the textual encoder even if an expressively powerful pre-trained Transformer encoder is applied. However, such problem only posts little effects on traditional graph embedding approach since they only consider the structure of the knowledge base.

Lexical Gap. Since the components of a knowledge base are human-curated, there is a lexical gap between the texts of a graph triple and natural language sentence from raw corpora; especially the Transformer encoder is pre-trained on large-scale raw corpora. Such lexical gap would lead to a performance drop on some certain relations. For instance, when dealing with a relation of "*derivationally-related-form*", the proposed model tends to assign a high ranking score to a triple in which the two entity are the same one and hence leads to errors. We find that, in both training and test phase, two entities at both ends of the relation frequently have the same textual knowledge but have subtly different meanings. For example, in the triple of "(ration, derivationally-related-form ration)", the first "*ration*" means "*a fixed portion that is allotted*" whereas the second one means "*restrict the consumption of a relatively scarce commodity*". This problem further deteriorates the entity ambiguity problem, and also has negative effects.

5 Related Work

Traditional Graph Embedding Approaches. To perform knowledge based completion such as link prediction, traditional graph embedding approaches aim at assigning distributed low-dimensional vectors to entities and relations. These approaches can be grouped into three classes according to the way to modeling the knowledge base. The approach in the first class aims at modeling a translation relationship in a graph triple, and thus defines a score function over the graph triple. The score function is built upon that, the geometric distance between $f(\boldsymbol{h}, \boldsymbol{r})$ and $\boldsymbol{t}$ should be short for a correct triple while long for a wrong triple. For example, TransE [1] proposes to use Euclidean distance and formulate the score function as $-||\boldsymbol{h} + \boldsymbol{r} - \boldsymbol{t}||$. It uses margin-based hinge loss to maximize the margin of the scores of correct triple and its corrupted one. To mine compositing rules in knowledge base, DistMult [31] update the formulation to $- < \boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t} >$. Moreover, to leverage the great features of complex vector, ComplEx [26] proposes to use complex vectors to represent the entities and relations, and use $-\mathrm{Re}(< \boldsymbol{h}, \boldsymbol{r}, \bar{\boldsymbol{t}} >)$ as the score function. More recently, RotatE [24] further exploits the complex vectors and propose applying relation can be viewed as a kind of rotate of the complex vector of the head entity. Such a method can handle the relations of symmetry, antisymmetry, inversion and composition in the meantime, and thus achieves state-of-art performance on link prediction. Rather than using simple geometric distance to model a graph triple, the approach in the second class employs CNNs to model a graph triple. For example, ConvE [5] applies a 2D-CNN to a reshaped concatenation of the embeddings of head entity and relation, and derives the plausibility score via a dot-product between the output from CNN and the embedding of tail entity. Similarly, ConvKB [15] straightforwardly apply a CNN to a concatenation of a triple's embeddings to produce a plausibility score of the triple. The approach in the third class aims at using graph convolutional network (GCN) to make the neural network deeper and thus enrich graph embeddings via learning structured information. R-GCN [20] proposes a relational graph convolutional network as an extension to GCN to handle the relational information.

Textual Encoding for Knowledge Base Completion. Instead of considering only the structured knowledge of the graph, textual encoding approaches [23,28–30] propose to incorporate the textual information with structured knowledge for knowledge base completion. For example, SSP [29] incorporates textual embeddings derived from entities' content with the symbolic ones and proposes a topic-related training loss in addition to the one based on the graph's structure. Nonetheless, there are two disadvantages regarding to these approaches. First, the head entity, relation and tail entity in a graph triple are encoded individually, despite the rich contextual features between an entity and relation. Second, only shallow 1D-CNN [8] or continuous bag-of-words (CBoW) is used to model the text. Recently, pre-trained language models, such as ELMo [17], BERT [6], RoBERTa [10] and GPT [18], have been proven effective in generating expressively powerful contextualized representations and shown promising performance when fine-tuning on downstream tasks. In addition, it is verified that the pre-trained language models have been partially equipped with commonsense structured knowledge [4,7]. Therefore, many attempts [2,12] have been made to apply the pre-trained Transformer encoder to commonsense knowledge base completion tasks. For example, [12] learns graph structure via GCN and pre-trained Transformer encoder to solve commonsense link prediction task. More related to this work, KG-BERT [32] leverages the pre-trained Transformer encoder to solve the knowledge base completion tasks over non-commonsense knowledge bases like Freebase and WordNet.

6 Conclusion and Discussion

In this paper, we employ a Siamese Transformer encoder operating on each graph triple to tackle link prediction tasks for knowledge base completion. Siamese architecture solves the combinatorial explosion issue in test phase and thus ensures a high efficiency of the proposed model. In addition, although a graph triple is split into two parts to suit the Siamese network, the contextual information across the entity and relation is still captured by the carefully designed model structure. Therefore, the proposed method is able to perform fast and effective link predictions. In the experiments, the proposed method reaches state-of-the-art or comparable performance on several link prediction datasets, with a superior efficiency compared to its baseline.

However, the proposed textual encoding approach inevitably struggles with entity ambiguity problem, and is a lack of structured information. Therefore, in the future we will focus on how to incorporate the semantic knowledge underlying texts with structured information from graph, e.g., via ensemble, pipeline model or heterogeneous embedding, and further boost the performance on knowledge base completion.

Acknowledgements We thank Guodong Long for his constructive and in-depth comments to this work.

References

1. Bordes A, Usunier N, García-Durán A, Weston J, Yakhnenko O. (2013) Translating embeddings for modeling multi-relational data. In: Burges CJC, Bottou L, Ghahramani Z, Weinberger KQ (eds) Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States, pp. 2787–2795. http://papers.nips.cc/paper/5071-translating-embeddings-for-modeling-multi-relational-data
2. Bosselut A, Rashkin H, Sap M, Malaviya C, Çelikyilmaz A, Choi Y. (2019) COMET: commonsense transformers for automatic knowledge graph construction. In: Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, pp. 4762–4779. https://doi.org/10.18653/v1/p19-1470
3. Chopra S, Hadsell R, LeCun Y (2005) Learning a similarity metric discriminatively, with application to face verification. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA, pp. 539–546. https://doi.org/10.1109/CVPR.2005.202
4. Davison J, Feldman J, Rush AM (2019) Commonsense knowledge mining from pretrained models. In: Inui K, Jiang J, Ng V, Wan X (eds) Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, pp. 1173–1178. Association for Computational Linguistics https://doi.org/10.18653/v1/D19-1109
5. Dettmers T, Minervini P, Stenetorp P, Riedel S (2018) Convolutional 2d knowledge graph embeddings. In: McIlraith SA, Weinberger KQ (eds) Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, pp. 1811–1818. AAAI Press. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17366
6. Devlin J, Chang M, Lee K, Toutanova K (2019) BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein J, Doran C, Solorio T (eds) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pp. 4171–4186. Association for Computational Linguistics. https://doi.org/10.18653/v1/n19-1423
7. Jastrzebski S, Bahdanau D, Hosseini S, Noukhovitch M, Bengio Y, Cheung JCK (2018) Commonsense mining as knowledge base completion? A study on the impact of novelty. CoRR http://arxiv.org/abs/1804.09259
8. Kim Y (2014) Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pp. 1746–1751. https://doi.org/10.3115/v1/d14-1181
9. Krompaß D, Baier S, Tresp V (2015) Type-constrained representation learning in knowledge graphs. In: Arenas M, Corcho Ó, Simperl E, Strohmaier M, d'Aquin M, Srinivas K, Groth PT, Dumontier M, Heflin J, Thirunarayan K, Staab S (eds) The Semantic Web - ISWC 2015-14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part I, *Lecture Notes in Computer Science*, vol. 9366, pp. 640–655. Springer. https://doi.org/10.1007/978-3-319-25007-6_37
10. Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V (2019) Roberta: A robustly optimized BERT pretraining approach. CoRR arXiv:1907.11692
11. Liu Z, Xiong C, Sun M, Liu Z (2018) Entity-duet neural ranking: Understanding the role of knowledge graph semantics in neural information retrieval. In: Gurevych I, Miyao Y (eds) Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers, pp. 2395–2405. Association for Computational Linguistics. https://www.aclweb.org/anthology/P18-1223/

12. Malaviya C, Bhagavatula C, Bosselut A, Choi Y (2019) Exploiting structural and semantic context for commonsense knowledge base completion. CoRR arXiv:1910.02915
13. Miller GA (1998) WordNet: An electronic lexical database. MIT press, Cambridge
14. Nathani D, Chauhan J, Sharma C, Kaul M (2019) Learning attention-based embeddings for relation prediction in knowledge graphs. In: Korhonen A, Traum DR, Màrquez L (eds) Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, pp. 4710–4723. Association for Computational Linguistics. https://doi.org/10.18653/v1/p19-1466
15. Nguyen DQ, Nguyen TD, Nguyen DQ, Phung DQ (2018) A novel embedding model for knowledge base completion based on convolutional neural network. In: Walker MA, Ji H, Stent A (eds) Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers), pp 327–333. Association for Computational Linguistics. https://doi.org/10.18653/v1/n18-2053
16. Nguyen DQ, Vu T, Nguyen TD, Nguyen DQ, Phung DQ (2019) A capsule network-based embedding model for knowledge graph completion and search personalization. In: Burstein J, Doran C, Solorio T (eds) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pp. 2180–2189. Association for Computational Linguistics. https://doi.org/10.18653/v1/n19-1226
17. Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L (2018) Deep contextualized word representations. In: Walker MA, Ji H, Stent A (eds) Proceedings of the 2018 conference of the north american chapter of the association for computational linguistics: human language technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers), pp. 2227–2237. Association for Computational Linguistics. https://doi.org/10.18653/v1/n18-1202
18. Radford A, Narasimhan K, Salimans T, Sutskever I (2018) Improving language understanding by generative pre-training. URL https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/languageunderstandingpaper.pdf
19. Reimers N, Gurevych I (2019) Sentence-bert: Sentence embeddings using siamese bert-networks. In: Inui K, Jiang J, Ng V, Wan X (eds) Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, pp. 3980–3990. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1410
20. Schlichtkrull MS, Kipf TN, Bloem P, van den Berg R, Titov I, Welling M (2018) Modeling relational data with graph convolutional networks. In: Gangemi A, Navigli R, Vidal M, Hitzler P, Troncy R, Hollink L, Tordai A, Alam M (eds) The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings, *Lecture Notes in Computer Science*, vol. 10843, pp. 593–607. Springer. https://doi.org/10.1007/978-3-319-93417-4_38
21. Shang C, Tang Y, Huang J, Bi J, He X, Zhou B (2019) End-to-end structure-aware convolutional networks for knowledge base completion. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019, pp. 3060–3067. AAAI Press. https://doi.org/10.1609/aaai.v33i01.33013060
22. Shen T, Geng X, Qin T, Guo D, Tang D, Duan N, Long G, Jiang D (2019) Multi-task learning for conversational question answering over a large-scale knowledge base. In: Inui K, Jiang J, Ng V, Wan X (eds) Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, pp. 2442–2451. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1248
23. Socher R, Chen D, Manning CD, Ng AY (2013) Reasoning with neural tensor networks for knowledge base completion. In: Burges CJC, Bottou L, Ghahramani Z, Weinberger KQ (eds) Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States, pp. 926–934. http://papers.nips.cc/paper/5028-reasoning-with-neural-tensor-networks-for-knowledge-base-completion
24. Sun Z, Deng Z, Nie J, Tang J (2019) Rotate: Knowledge graph embedding by relational rotation in complex space. In: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net. https://openreview.net/forum?id=HkgEQnRqYQ

25. Toutanova K, Chen D, Pantel P, Poon H, Choudhury P, Gamon M (2015) Representing text for joint embedding of text and knowledge bases. In: Màrquez L, Callison-Burch C, Su J, Pighin D, Marton Y (eds) Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015, pp. 1499–1509. The Association for Computational Linguistics. https://doi.org/10.18653/v1/d15-1174
26. Trouillon T, Welbl J, Riedel S, Gaussier É, Bouchard G (2016) Complex embeddings for simple link prediction. In: Balcan M, Weinberger KQ (eds) Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016, *JMLR Workshop and Conference Proceedings*, vol. 48, pp. 2071–2080. JMLR.org. http://proceedings.mlr.press/v48/trouillon16.html
27. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. In: Guyon I, von Luxburg U, Bengio S, Wallach HM, Fergus R, Vishwanathan SVN, Garnett R (eds) Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA, pp. 5998–6008. http://papers.nips.cc/paper/7181-attention-is-all-you-need
28. Wang Z, Zhang J, Feng J, Chen Z (2014) Knowledge graph and text jointly embedding. In: Moschitti A, Pang B, Daelemans W (eds) Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pp. 1591–1601. ACL. https://doi.org/10.3115/v1/d14-1167
29. Xiao H, Huang M, Meng L, Zhu X (2017) SSP: semantic space projection for knowledge graph embedding with text descriptions. In: Singh SP, Markovitch S (eds) Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA, pp. 3104–3110. AAAI Press. http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14306
30. Xie R, Liu Z, Jia J, Luan H, Sun M (2016) Representation learning of knowledge graphs with entity descriptions. In: Schuurmans D, Wellman MP (eds) Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA, pp. 2659–2665. AAAI Press. http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12216
31. Yang B, Yih W, He X, Gao J, Deng L (2015) Embedding entities and relations for learning and inference in knowledge bases. In: Bengio Y, LeCun Y (eds) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings. http://arxiv.org/abs/1412.6575
32. Yao L, Mao C, Luo Y (2019) KG-BERT: BERT for knowledge graph completion. CoRR arXiv:1909.03193
33. Yu J, Kuang Z, Zhang B, Zhang W, Lin D, Fan J (2018) Leveraging content sensitiveness and user trustworthiness to recommend fine-grained privacy settings for social image sharing. IEEE Trans Inf Forensics Secur 13(5):1317–1332. https://doi.org/10.1109/TIFS.2017.2787986
34. Yu T, Yu J, Yu Z, Tao D (2020) Compositional attention networks with two-stream fusion for video question answering. IEEE Trans Image Process 29:1204–1218. https://doi.org/10.1109/TIP.2019.2940677
35. Zhang S, Tay Y, Yao L, Liu Q (2019) Quaternion knowledge graph embeddings. In: Wallach HM, Larochelle H, Beygelzimer A, d'Alché-Buc F, Fox EB, Garnett R (eds) Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada, pp. 2731–2741. http://papers.nips.cc/paper/8541-quaternion-knowledge-graph-embeddings

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.