Knowledge-Infused Pre-trained Models for KG Completion

Han Yu, Rong Jiang, Bin Zhou, and Aiping Li(✉)

College of Computer, National University of Defense Technology, Changsha, China
{yuhan17,jiangrong,liaiping}@nudt.edu.cn

Abstract. Knowledge graphs (KG) are the basis for many artificial intelligence applications but still suffer from incompleteness. In this paper, we introduce a novel method for KG completion task by knowledge-infused pre-trained language models. We represent each triple in the KG as textual sequences and transform the KG completion task into a sentence classification task that fits the input of the language model. Our KG completion framework based on the knowledge-infused pre-trained language model which can capture both linguistic information and factual knowledge to compute the plausible of the triples. Experiments show that our method achieves better results than previous state-of-the-art on multiple benchmark datasets.

Keywords: Knowledge graph completion · Link prediction · Relation prediction · Pre-trained language model

1 Introduction

Knowledge graphs (KG) are structured knowledge bases, where facts are represented in the form of entities and relations. The entities are the nodes of the knowledge graph, and the relations are the edge between entities. Each edge and connected nodes form a triple $(head\ entity, relation, tail\ entity)$, indicating the relationship between entities, e.g.., $(Mark\_Twain, is\_a, writer)$. KG can be the basis for many applications: semantic search, recommendation, question answering, and data integration, etc. [11]. However, even large knowledge graphs such as FreeBase [2], YAGO [28], and WordNet [17], are still far from being complete, that is, missing relations or entities in the graphs [36]. This problem prompt the KG completion task which mainly includes link prediction and relation prediction to be proposed.

Many research efforts are devoted to KG completion, among them, knowledge graph embedding is an effective approach in which entities and edges are

The work described in this paper is partially supported by the National Key Research and Development Program of China (No. 2017YFB0802204, 2016QY03D0603, 2016QY03D0601, 2017YFB0803301, 2019QY1406), the Key R&D Program of Guangdong Province (No. 2019B010136003), and the National Natural Science Foundation of China (No. 61732004, 61732022, 61672020).

Z. Huang et al. (Eds.): WISE 2020, LNCS 12342, pp. 273–285, 2020.
https://doi.org/10.1007/978-3-030-62005-9_20

represented by embedding vectors. The embedding methods that use only knowledge graph structure information are often suffer from the sparsity of KG [14]. Therefore, Some recent studies incorporate extra text information to enrich knowledge representation [27,38,40]. These methods encode extra information as a unified word embeddings representation and cannot express the contextual information of the words in different contexts. For instance, in the two triples $(Mark\_Twain, is\_a, writer)$ and $(Mark\_Twain, born\_in, America)$, the same words in the description of $Mark\_Twain$ should have different importance weights related to the two relations is_a and $born\_in$. Besides, sufficient semantic and syntactic information cannot be learned by the small text of these methods such as the entity description.

Recently, BERT [6] and its various variants XLNet [42], RoBERTa [16], and ALBERT [12] have achieved great success in the field of natural language processing (NLP). These methods pre-trained with a large amount of unlabeled corpus and achieve state-of-the-art performance on several downstream NLP tasks by simply fine-tuning all pre-trained parameters. BERT can capture rich linguistic knowledge in pre-trained. BERT-based models have already effectively applied to various applications of NLP, such as question answering, reading comprehension, relationship extraction, dialogue generation, and it is also used in the KG completion [43]. However, some inference tasks require not only linguistic knowledge but also factual knowledge. To alleviate this problem, ERNIE [45], KnowBERT [21], K-BERT [15] and K-Adapter [34] inject knowledge into the language model.

For KG completion task, factual knowledge is particularly important for inferring relations between entities. Consider our previous example $(Mark\_Twain, born\_in, America)$. Given the head entity $Mark\_Twain$ and the relation $born\_in$. [22] suggest that the pre-trained language models may infer the tail entity $America$ by the surface form of entity name, because $Mark\_Twain$ to be a common American name. But when a person with an Italian name was born in the America, we need to use factual knowledge to reason the tail entity. In this study, we propose a novel method for KG completion using knowledge-infused pre-trained language models. For each triple, we span the entities and relation into text sequences and convert the completion of the knowledge graph into sequence classification problems. Then, we fine-tune the knowledge-infused BERT on these sequences to predict the plausibility of triples. The contributions of our paper are as follows:

- We propose a novel method for KG completion using knowledge-infused pre-trained language models. And to the best of our knowledge, this is the first study to use a knowledge-infused pre-trained language model for KG completion.
- Evaluating results on several benchmark datasets show that our method can achieve state-of-the-art performance in KG completion tasks.

2 Related Work

KG Embedding KG embedding methods can be classified into translational distance models and semantic matching models based on different scoring functions [33]. The representative translational distance models are TransE [3] and its extensions include TransH [35], TransD [9], etc. These models use distance-based scoring functions to evaluate the plausibility of a triple. The semantic matching models employ similarity-based scoring functions, and the typical models are RESCAL [20] and DistMult [44]. In addition, the convolutional neural networks (CNN) based methods ConvKB [18], ConvE [5], R-GCN [24] show promising results for KG completion.

The above methods only use structure information for KG completion, while some methods introduce external information to improve the performance [33]. NTN [27] represents the entities by word embeddings that are learned from the external corpus. DKRL [40] encodes the entity descriptions and learn embeddings with both triples and descriptions. SSP [37] learns the topic and KG embeddings together by characterizing the correlation between fact triples and text descriptions. Through external information, the effectiveness of these models can be improved, but these methods use the same word embedding weights to represent the entities and relations in different triples which would have different meanings.

To alleviate the above problems, TEKE [24] assigns different word embeddings to the relation in different triples. AATE [1] enhances representations by exploiting the entity descriptions and triple specific relation mention, then uses the mutual attention mechanism to learn more accurate textual representations. These methods can handle the semantic variety of entities and relations in distinct triples, but the ability of textual representation is limited by the small corpus such as entity descriptions. Compared with these methods, KG-BERT [43] can capture rich linguistic information via pre-trained language models. But they lack the factual knowledge information to grasp the relationship between entities which is important for KG completion tasks. Our method uses knowledge-infused language models to solve this problem.

Pre-trained Language Model Pre-trained language representation models can be divided into feature-based and fine-tuning methods. Feature-based methods only pre-trained word embedding parameters while fine-tuning methods learn the parameters of the pre-trained model architecture. Through fine-tuning, the pre-trained model can be applied in downstream tasks with few parameters need to be learned scratch. The representative fine-tuning method BERT achieves state-of-the-art results for various NLP tasks. Currently, BERT-based models are explored in fields such as question answering [8,41], reading comprehension [46], relation extraction [26], text classification [23], etc. And it also used in KG completion [43]. Though BERT can capture rich semantic information, but ignore the incorporation of knowledge information. Therefore, some works [21,34,45] injecting extra knowledge information into pre-trained language representation. In this study, we take a knowledge-infused pre-trained language model as the framework and fine-tune on the KG completion task.

3 Methodology

3.1 Knowledge-Infused BERT

BERT is a pre-trained language model built on the multi-layer bidirectional Transformer encoder. And it applied to downstream tasks through two steps of pre-training and fine-tuning. For pre-training, BERT is trained in a self-supervised way, and it trained with large corpus data (3,300 M words from BooksCorpus and English Wikipedia). For fine-tuning, BERT is initialized by the pre-trained parameters, and use labeled data from downstream tasks (such as sentence classification, question answering, etc.) to fine-tune all parameters. BERT can obtain rich contextual semantic and syntactic information through pre-training. Knowledge embedding methods (such as TransE [3]) which vectorize the structured KG can learn the knowledge information of entities and relations. To take full advantage of the contextual language representation of the pre-trained BERT and the factual knowledge of entities and relations in the KG, we apply Knowledge-infused BERT for KG completion.

We use ERNIE [45] as the knowledge-infused language model which consists of two encoders, T-Encoder and K-Encoder, to construct our framework. T-Encoder is responsible to capture basic lexical and syntactic information from the input tokens, and K-Encoder is to integrate extra factual knowledge information into textual information. The structure of T-Encoder is the same as BERT, which consists of multi-layer self-attention Transformer. Given the token sequence $\{Tok_1, ..., Tok_n\}$, T-Encoder generates semantic and syntactic embedding as follows,

$$\{T_1, ..., T_n\} = T{-}Encoder(\{Tok_1, ..., Tok_n\}). \tag{1}$$

Where $\{T_1, ..., T_n\}$ is the output embedding. K-Encoder is composed of stacked aggregators, which is similar to Transformer in structure that consists of multi-head self-attentions and infusion layer. The input of the K-Encoder is the output embedding of T-Encoder $\{T_1, ..., T_n\}$ and the entity embeddings $\{Ent_1, ..., Ent_n\}$ which is pre-trained by KG embedding methods. Then the K-Encoder injects the knowledge into language representation,

$$\{E_1, ..., E_n\} = K{-}Encoder(\{T_1, ..., T_n\}, \{Ent_1, ..., Ent_n\}). \tag{2}$$

Where $\{E_1, ..., E_n\}$ is the final output embedding. With K-Encoder, the heterogeneous information of semantic and syntactic information and factual knowledge can be integrated into a unified vector space. As T-Encoder and K-Encoder is identical to its implementation in ERNIE, readers can refer [45] for a more detailed description of the model.

3.2 KG Completion Framework

KG consists of structured entities and relationships. We define h as the head entity, r as the relationship, t as the tail entity, and (h, r, t) as the triple. We

fine-tune pre-trained knowledge-infused BERT for KG completion. When pre-training original knowledge-infused BERT, the input of the T-Encoder is continuous text or word sequence. Therefore, we turn the entity and relation which are their names or descriptions, into a sequence form as the input of the T-Encoder. And take the pre-trained KG embedding of the entity as the input of the K-Encoder.

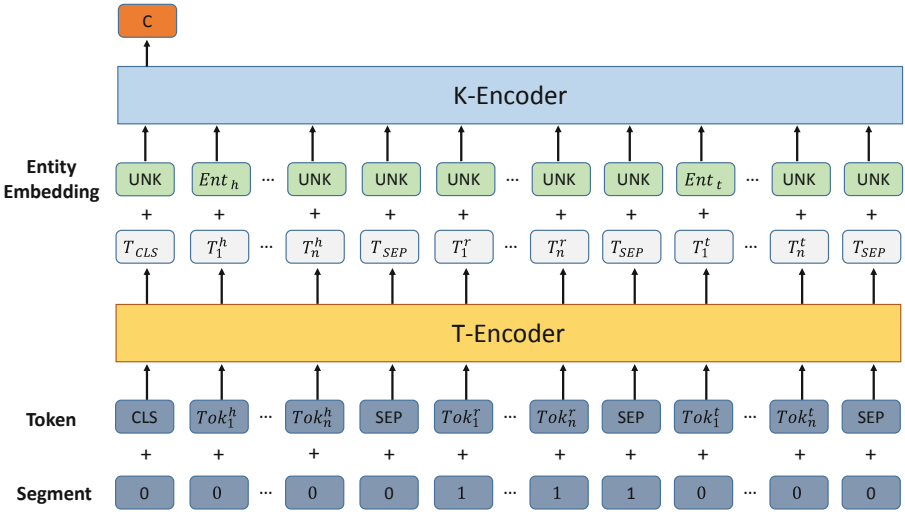

Fig. 1. The architecture of our framework. The input embeddings of K-Encoder are the sum of the token, segment and the default position embeddings of BERT. The input of K-Encoder are the sum of the output embeddings of T-Encoder and entity embedding sequence.

Link Prediction. The architecture of our framework for predicting the plausibility of a triple is shown in Fig. 1. For triple classification and link prediction task, the first input token of the T-Encoder is a special classification token [CLS]. The head entity is represented as a sequence of tokens $\{Tok_1^h, ..., Tok_n^h\}$, the relation is represent as $\{Tok_1^r, ..., Tok_n^r\}$ and the tail entity is represent as $\{Tok_1^t, ..., Tok_n^t\}$. The sequence of entity tokens and relation tokens are separated by a special token [SEP]. We then concatenate three token sequence to construct the input token sequence. For a given input token, its input representation of T-Encoder is constructed by summing the corresponding token, segment, and position embeddings. We set head entity and tail entity to the same segment embedding, and relation to another different segment embedding. We use the default position embedding of BERT that all token sequences in same location has the same position embedding. For K-Encoder, the input is the output embeddings of T-Encoder sums the head and tail entity embeddings $\{Ent_h, Ent_t\}$. We set the special token [UNK] as the first token of the entity

embedding sequence. If the entity is represented by name, the second token is the pre-trained knowledge embedding of the entity, and the remaining positions are filled with [UNK] to the same length as the T-Encoder input. If the entity is represented by a entity description, we first mark the entities in the description, then set the first token of the marked entity with knowledge embedding, and fill the rest with [UNK].

Firstly, the input representations are fed into the T-Encoder which is a multi-layer bidirectional Transformer encoder. And then the output embedding of T-Encoder and the entity embedding sequence are fed into the K-Encoder together. We use the first final hidden state of K-Encoder, which is corresponding to the [CLS] token, as the aggregate sequence representation for computing classification score. Given the hidden state, we introduce a classification layer to compute the triple scores,

$$s = sigmoid(CW). \tag{3}$$

Where C is the final hidden state aligned with the [CLS] token, $W \in \mathbb{R}^{H \times 2}$ is the parameters of the classification layer, H is the hidden state size.

We take the original triple in the knowledge graph as the positive triple set D^+, and define the negative triple set as D^-. For fine-tuning the model parameters, we minimize the following binary cross,

$$L = -\sum_{D^+ \cup D^-} (ylog(s) + (1-y)log(s)). \tag{4}$$

Where y is the label indicating that the triple is negative or positive. During fine-tuning, the pre-trained parameter weights and new weights W can be updated via gradient descent.

Relation Prediction. The framework of relation prediction is roughly the same as link prediction, except that there are no relation tokens in the token sequence. The architecture of relation prediction task is shown in Fig 2. We construct the token sequence composed by head entity tokens and tail entity tokens, without relation tokens. The head entity and tail entity tokens have different segment embeddings. We set the special token [CLS] as the first input token, and separate the head entity and tail entity with [SEP]. We also use the final hidden state C corresponding to [CLS] as the representation of the two entities. The scoring function for predicting relation is:

$$s' = sigmoid(CW'). \tag{5}$$

Where $W' \in \mathbb{R}^{H \times R}$ is the parameters of the classification layer for predicting relation, R is the number of relations in a knowledge graph. We minimize the cross-entropy loss to fine-tuning the model:

$$L' = -\sum_{D^+} \sum_{i=1}^{R} y'_i log(s'_i), \tag{6}$$

where y_i is the relation indicator for the triple.

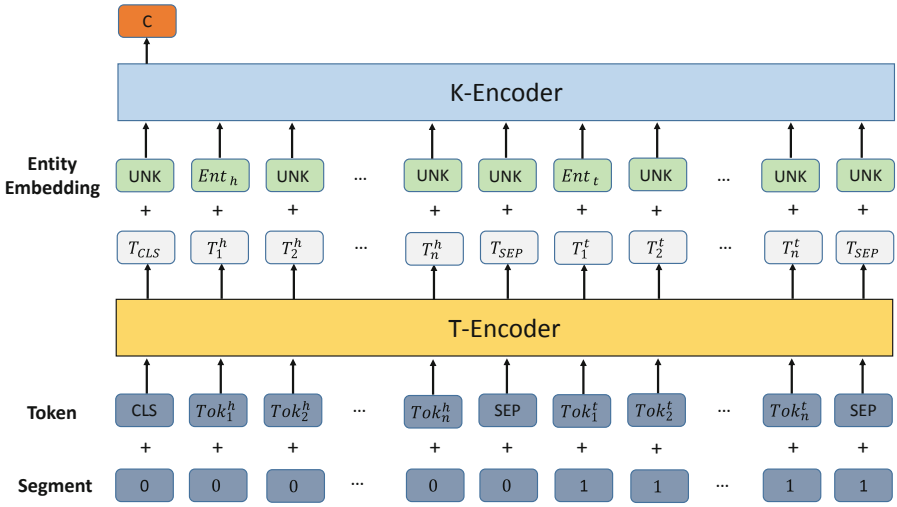

Fig. 2. The framework of relation prediction. The input embeddings of K-Encoder are the sum of the token, segment and the default position embeddings of BERT. The input of K-Encoder are the sum of the output embeddings of T-Encoder and entity embedding sequence.

Table 1. The statistics of datasets. Number of entities, relations, and observed triples in each split for benchmarks.

Dataset	Entities	Relations	Train	Dev	Test
WN11	38,696	11	112,581	2,609	10,544
FB13	75,043	13	316,232	5,908	23,733
WN18RR	40,943	11	86,835	3,034	3,134
FB15K	14,951	1,345	483,142	50,000	59,071
FB15k-237	14,541	237	272,115	17,535	20,466
UMLS	135	46	5,216	652	661

4 Experiments

In this section we evaluate our KG completion framework on three experimental tasks, triple classification, link prediction and relation prediction.

Datasets. We evaluate our experiments on six widely used benchmark KG datasets: WN11, FB13 [27], FB15K [3], WN18RR, FB15k-237 and UMLS [5]. Table 1 provides statistics of all datasets used in our experiment. WN11 and WN18RR are the subsets of WordNet which is a large lexical knowledge graph of English. FB15K and FB15k-237 are the subsets of Freebase which is a large knowledge graph about general facts. As noted by [30], WN18 and FB15k

are information leaked because they contain many reversible relation, while WN18RR and FB15k237 are created to not suffer from this reversible relation problem in WN18 and FB15k, for which the KG completion task is more realistic. UMLS is a medical semantic network containing semantic types (entities) and semantic relationships. We use the test sets of WN11 and FB13 which contain positive and negative triplets to evaluate triple classification. And we use test set of WN18RR, FB15K, FB15k-237 and UMLS which only contain correct triples to perform link prediction and relation prediction.

Baselines. We compare our framework with multiple state-of-the-art knowledge embedding methods including transition-based models TransE [3] and its extensions TransH [35], TransD [9], TransR [14], TransG [39], TranSparse [10] and PTransE [13], DistMult [44] which only used structural information in knowledge graphs. The neural tensor network NTN [27] and ProjE [25]. CNN-based models: ConvKB [18], ConvE [5] and R-GCN [24]. Textual information infused methods: TEKE [24], DKRL [40], SSP [37], AATE [1]. KG embeddings with entity hierarchical types TKRL [40]. Contextualized KG embeddings DOLORES [32]. Complex-valued KG embeddings ComplEx [31] and RotatE [29]. Adversarial learning framework KBGAN [4]. BERT-based framework KG-BERT [43].

Settings. We use pre-trained ERNIE [45] model with 6 layers of T-Encoder and 6 layers of K-Encoder. We denote the hidden dimension of token embeddings is $H_w = 768$, and hidden dimension of entity embeddings is $H_e = 100$. We set 12 self-attention heads for token embeddings and 4 self-attention heads for entity embeddings. In our framework, we set the hyper-parameters of batch size to 32, learning rate to 5e-5, and the dropout rate to 0.1. We fine-tune our framework with Adam implemented in BERT. We tuned 3 epochs for triple classification, 5 for link prediction and 20 for relation prediction. For triple classification training, we sample 1 negative triple for a positive triple. For link prediction training, we sample 5 negative triples for a positive triple. For relation prediction training, we only use positive triple.

To capture the relationships among entities in the pre-trained language model, we add relation classification task as the pre-training processes. We use a subset of T-REx [7] which is a large scale alignment dataset to pre-train the model to classify relation labels of given entity pairs based on context. For the UMLS dataset, since there lacks sufficient medical knowledge in the pre-trained know-
ledge-infused language model, we pre-train the knowledge-infuse framework with PubMed abstracts, PubMed Central full-text papers and the entity embeddings of UMLS.

Triple Classification. Triple classification is to infer whether a triple is the correct triple or not. Table 2 shows the results of various models performing triple classification on WN11 and FB13 datasets. We ran our models 3 times and average the accuracy of each time as the final result. We can see that the BERT-based methods have a large improvement over the results of other baseline models. Our framework performance better than KG-BERT, proving the effectiveness of our

Table 2. Results on triple classification for different embedding methods.

Method	WN11	FB13	avg
NTN	86.2	90.0	88.1
TransE	75.9	81.5	78.7
TransH	78.8	83.3	81.1
TransR	85.9	82.5	84.2
TransD	86.4	89.1	87.8
TEKE	86.1	84.2	85.2
TransG	87.4	87.3	87.4
TranSparse-S	86.4	88.2	87.3
DistMult	87.1	86.2	86.7
DistMult-HRS	88.9	89.0	89.0
AATE	88.0	87.2	87.6
ConvKB	87.6	88.8	88.2
DOLORES	87.5	89.3	88.4
KG-BERT	93.5	90.4	91.9
Ours	**93.5**	**90.5**	**92.0**

method. Analysis of the results, the effectiveness of the BERT-based methods have two main fold: First, the baseline models no matter uses structural information or extra text information not the utilization of rich language patterns, and the BERT-based methods can obtain rich linguistic patterns information from a large amount of corpus through pre-training. Second, entities connected by different relationships have different meanings, and words have different semantics in the corpus according to different contexts. BERT-based methods can make full use of the learned contextual information in the triple classification process. In addition, the BERT-based methods have achieved a greater improvement on WN11, that may because WordNet is a linguistic knowledge graph, which is closer to the language model. Our model obtained more improvements in the FB13 dataset. The reason for the improvement is that the original BERT model learns more about the semantic association between tokens than the knowledge between entities. The knowledge-infused BERT can use the extra factual knowledge between entities by injecting knowledge into the language model. Note that, if we do not use the entity embeddings as the input of K-Encoder (that is, replace all *Ent* with [UNK]), the performance of our framework will decline compared with KG-BERT. This is because knowledge-infused language model forgets part of the linguistic information during the pre-training process.

Link Prediction. Link prediction task predicts the head entity given the relation and tail entity, or predicts the tail entity given the relation and head entity. We following the protocol of [19] that only report results under the filtered setting [3] which removes all corrupted triples appeared in training data and testing

data before getting the ranking lists. We use two common metric Mean Rank (MR) and Hit@10 to evaluate the performance of models. A lower MR is better while a higher Hits@10 is better.

Table 3 represents link prediction performance of various models. We take the results of baseline models from the original papers. We observe that BERT-based methods get lower MR than other baseline models. It because pre-trained BERT can capture the semantic relatedness of entity and relation sentences to avoid very high ranks. BERT has not learned the knowledge graph structural information between entities, so BERT-based methods not achieve higher Hit@10 than some state-of-the-art models. The knowledge-infused BERT injects entity

Table 3. Link prediction results on WN18RR, FB15k-237 and UMLS datasets.

Method	WN18RR		FB15k-237		UMLS	
	MR	Hit@10	MR	Hit@10	MR	Hit@10
TransE	2365	50.5	223	47.4	1.84	98.9
TransH	2524	50.3	255	48.6	1.80	**99.5**
TransR	3166	50.7	237	51.1	1.81	99.4
TransD	2768	50.7	246	48.4	1.71	99.3
DistMult	3704	47.7	411	41.9	5.52	84.6
ComplEx	3921	48.3	508	43.4	2.59	96.7
ConvE	5277	48	246	49.1	–	–
ConvKB	2554	52.5	257	51.7	–	–
R-GCN	–	–	–	41.7	–	–
KBGAN	–	48.1	–	45.8	–	–
RotatE	3340	**57.1**	177	**53.3**	–	–
KG-BERT	97	52.4	153	42.0	1.47	99.0
Ours	**96**	52.7	**149**	43.1	**1.45**	99.3

Table 4. Relation prediction results on FB15k dataset.

Method	Mean rank	Hit@1
TransE	2.5	84.3
TransR	2.1	91.6
DKRL	2.0	90.8
TKRL	1.7	92.8
PTransE	1.2	93.6
SSP	1.2	–
ProjE	1.2	95.7
KG-BERT	1.2	96.0
Ours	**1.2**	**96.2**

knowledge into the language model to better learn the relationship between entities during the pre-training, therefore, our framework get higher Hit@10 than KG-BERT. Without the relation classification task added in the pre-training, our framework has lower performance than KG-BERT due to the lack of learned relationship between entities and the loss of some semantic information.

Relation Prediction. Relation prediction task is to predict relation given the head and the tail entities. The procedure is similar to link prediction and we evaluate the models using MR and Hits@1 with filtered setting. Table 4 shows the results of relation prediction task on FB15K. We note that our framework also shows promising results and achieves the highest Hits@1 so far. The relation prediction task is analogous to sentence pair classification in BERT fine-tuning and can also benefit from BERT pre-training. Knowledge-infused BERT not only learns the semantic representation of entities, but also the knowledge representation between entities, so we get better results than KG-BERT.

5 Conclusion

In this paper, we presented a novel method for KG completion, and outperforming existing methods on multiple benchmark datasets. Our method use knowledge-infused pre-trained language model and turn the triples in the KG into token sequences. Then transform the KG completion task into sentence classification. The experiments demonstrate that our method outperforms state-of-the-art results on several benchmark datasets. In the future, it is promising to exploit incorporate more KG structured information, and inject factual knowledge to compute the plausible of the triples without re-training.

References

1. An, B., Chen, B., Han, X., Sun, L.: Accurate text-enhanced knowledge graph representation learning. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp. 745–755 (2018)
2. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge, pp. 1247–1250 (2008)
3. Bordes, A., Usunier, N., Garciaduran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data, pp. 2787–2795 (2013)
4. Cai, L., Wang, W.Y.: Kbgan: adversarial learning for knowledge graph embeddings. arXiv preprint arXiv:1711.04071 (2017)
5. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2D knowledge graph embeddings. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
6. Devlin, J., Chang, M., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv: Computation and Language (2018)

7. Elsahar, H., et al.: A large scale alignment of natural language with knowledge base triples, T-rex (2018)
8. Godbole, A., Kavarthapu, D., Das, R., Gong, Z., McCallum, A.: Multi-step entity-centric information retrieval for multi-hop question answering
9. Ji, G., He, S., Xu, L., Liu, K., Zhao, J.: Knowledge graph embedding via dynamic mapping matrix. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 687–696 (2015)
10. Ji, G., Liu, K., He, S., Zhao, J.: Knowledge graph completion with adaptive sparse transfer matrix. In: Thirtieth AAAI Conference on Artificial Intelligence (2016)
11. Jin, J., Luo, J., Khemmarat, S., Dong, F., Gao, L.: GSTAR: an efficient framework for answering top-k star queries on billion-node knowledge graphs. World Wide Web **22**(4), 1611–1638 (2019)
12. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: Albert: a lite bert for self-supervised learning of language representations
13. Lin, Y., Liu, Z., Luan, H., Sun, M., Rao, S., Liu, S.: Modeling relation paths for representation learning of knowledge bases. arXiv preprint arXiv:1506.00379 (2015)
14. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion, pp. 2181–2187 (2015)
15. Liu, W., et al.: K-Bert: enabling language representation with knowledge graph. arXiv preprint arXiv:1909.07606 (2019)
16. Liu, Y., et al.: Roberta: A robustly optimized Bert pretraining approach
17. Miller, G.A.: Wordnet: a lexical database for English. Commun. ACM **38**(11), 39–41 (1995)
18. Nguyen, D.Q., Nguyen, D.Q., Nguyen, T.D., Phung, D.: A convolutional neural network-based model for knowledge base completion and its application to search personalization. Semant. Web **10**(5), 947–960 (2019)
19. Nguyen, D.Q., Nguyen, T.D., Nguyen, D.Q., Phung, D.: A novel embedding model for knowledge base completion based on 22 convolutional neural network
20. Nickel, M., Tresp, V., Kriegel, H.-P.: A three-way model for collective learning on multi-relational data. ICML **11**, 809–816 (2011)
21. Peters, M.E., et al.: Knowledge enhanced contextual word representations, pp. 43–54 (2019)
22. Poerner, N., Waltinger, U., Schütze, H.: Bert is not a knowledge base (yet): Factual knowledge vs. name-based reasoning in unsupervised QA
23. Reimers, N., Schiller, B., Beck, T., Daxenberger, J., Gurevych, I.: Classification and clustering of arguments with contextualized word embeddings. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (2019)
24. Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: Gangemi, A., et al. (eds.) ESWC 2018. LNCS, vol. 10843, pp. 593–607. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93417-4_38
25. Shi, B., Weninger, T.: Proje: embedding projection for knowledge graph completion. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
26. Soares, L.B., FitzGerald, N., Ling, J., Kwiatkowski, T.: Matching the blanks: distributional similarity for relation learning
27. Socher, R., Chen, D., Manning, C.D., Ng, A.Y.: Reasoning with neural tensor networks for knowledge base completion, pp. 926–934 (2013)
28. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge, pp. 697–706 (2007)

29. Sun, Z., Deng, Z.-H., Nie, J.-Y., Tang, J.: Rotate: knowledge graph embedding by relational rotation in complex space. arXiv preprint arXiv:1902.10197 (2019)
30. Toutanova, K., Chen, D.: Observed versus latent features for knowledge base and text inference. In: Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality, pp. 57–66 (2015)
31. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction (2016)
32. Wang, H., Kulkarni, V., Wang, W.Y.: Dolores: deep contextualized knowledge graph embeddings. arXiv preprint arXiv:1811.00147 (2018)
33. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: a survey of approaches and applications. IEEE Trans. Knowl. Data Eng. **29**(12), 2724–2743 (2017)
34. Wang, R., et al.: K-adapter: infusing knowledge into pre-trained models with adapters. arXiv: Computation and Language (2020)
35. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Twenty-Eighth AAAI Conference on Artificial Intelligence (2014)
36. Wu, T., et al.: Knowledge graph construction from multiple online encyclopedias. World Wide Web, pp. 1–28 (2019)
37. Xiao, H., Huang, M., Meng, L., Zhu, X.: SSP: semantic space projection for knowledge graph embedding with text descriptions. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
38. Xiao, H., Huang, M., Zhu, X.: SSP: semantic space projection for knowledge graph embedding with text descriptions. arXiv: Computation and Language (2016)
39. Xiao, H., Huang, M., Zhu, X.: Transg: a generative model for knowledge graph embedding. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 2316–2325 (2016)
40. Xie, R., Liu, Z., Sun, M.: Representation learning of knowledge graphs with hierarchical types, pp. 2965–2971 (2016)
41. Yang, W., et al.: End-to-end open-domain question answering with Bertserini
42. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., Le, Q.V.: XlNet: generalized autoregressive pretraining for language understanding
43. Yao, L., Mao, C., Luo, Y.: KG-Bert: Bert for knowledge graph completion
44. Zhang, Z., Zhuang, F., Qu, M., Lin, F., He, Q.: Knowledge graph embedding with hierarchical relation structure. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 3198–3207 (2018)
45. Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., Liu, Q.: Ernie: enhanced language representation with informative entities, pp. 1441–1451 (2019)
46. Zhu, H., Dong, L., Wei, F., Wang, W., Qin, B., Liu, T.: Learning to ask unanswerable questions for machine reading comprehension