Investigating Knowlege Graph Completion with Pre-trained Language model and Graph Neural Networks

Amit Roy, Taisuke Mori
Computer Science, Purdue University
`{roy206,tmori}@purdue.edu`

Abstract

In recent years, two lines of research in deep learning are gaining lots of interest namely Large Language Models and Graph Neural Networks. The earlier one has shown reasonable performance in different types of NLP tasks and capturing the structural relationship with the graph convolution operation is the key reason behind the success of GNNs. Predicting links in knowledge graphs is a very important task for knowledge completion and information retrieval. Considering a knowledge graph has text data in their entities and relations and also provides the structural information in the form of a triplet, some state-of-the-art approaches have *separately* employed BERT-like large language models and others have adopted GNNs to predict links in a knowledge graph. In this study, we aim to investigate how the combination of embeddings from the Large Language Models and a graph convolution network can improve the link prediction task. We performed experiments on three real-world knowledge graph datasets and demonstrated our results [1], [2].

1 Introduction

Knowledge graphs, which represent relation between entities, have a wide variety of applications such as question answering and information retrieval (Yao and Van Durme, 2014; Dalton et al., 2014). Given the fact that most knowledge bases built in real-world do not cover all information, providing missing relations via link prediction between entities are a crucial process to fully utilize those knowledge bases.

In this decade, deep learning is advancing rapidly, where Large Language Models e.g. BERT, perform well on varieties of NLP tasks, and Graph Neural Networks (GNN) with their graph convolution operation can produce expressive representations suitable for different downstream tasks. Although both language models and GNNs are separately applied for knowledge graph completion (Wang et al., 2019, 2022; Schlichtkrull et al., 2018; Yao et al., 2019), there are few works that take advantage of both techniques. Specifically, in the language model-based approach, the graph structure is not explicitly used while in GNN-based approaches, models do not exploit textual information of entities such as entity names and descriptions. Without utilizing the related information from the edges the large language model can not fully utilize the structure information of the knowledge graph. At the same time, avoiding the text information from the embedding can not allow the GCN-based model namely RGCN to obtain a useful representation for the downstream task.

In this project, we aim to investigate how we can improve the performance of knowledge graph completion tasks by combining language models with GNNs. We would like to utilize the representations obtained from the pre-trained language model and feed them to GNN-based models for knowledge graph completion and observe how the change of the input embeddings changes the performance.

1.1 Problem statement

Consider a directed labeled graph $G = (\mathcal{V}, \mathcal{E}, \mathcal{R})$ where for entities $s, o \in \mathcal{V}$ and relations $r \in \mathcal{R}$, $(s, r, o) \in \mathcal{E}$ denotes the relationship $(subject, relation, object)$. Our goal is, given a graph $\hat{G} = (\mathcal{V}, \hat{\mathcal{E}}(\subset \mathcal{E}), \hat{\mathcal{R}}(\subset \mathcal{R}))$, to learn a scoring function $f : \mathcal{V} \times \mathcal{R} \times \mathcal{V} \to \mathbb{R}$ that gives a high score to a true triplet (s, r, o) by minimizing the cross entropy where positive examples are $(s, r, o) \in \hat{G}$ and negative examples are obtained by replacing either subject or object of the (s, r, o) with a randomly chosen entity. We assume a pre-trained BERT model and Wikipedia corpus are

[1] Code available at `https://github.com/AmitRoy7781/BERT-RGCN`

[2] Slides available here

available besides the triplets.

2 Method

2.1 R-GCN

Relational Graph Convolutional Network (R-GCN) is developed by (Schlichtkrull et al., 2018) as an extension of a classic graph convolutional network (Kipf and Welling, 2016) to model large scale relational data. To encode a multi-relational graph structure into node representations, each layer of R-GCN aggregates from the previous layer the representations of the nodes with respect to all relations:

$$h_i^{(l+1)} = \sigma\left(\sum_{r\in\mathcal{R}}\sum_{j\in\mathcal{N}_i^r}\frac{1}{c_{i,r}}W_r^{(l)}h_j^{(l)} + W_0^{(l)}h_i^{(l)}\right)$$

where $h_i^{(l)}$ denotes the representation of ith node in lth layer, N_i^r denotes the set of neighboring nodes of ith node with respect to relation r, $W_r^{(l)}$ and $W_0^{(l)}$ are weight matrices and σ is an element-wise activation function. $c_{i,r}$ is a normalization constant, for which $|N_i^r|$ is commonly chosen. Since data has a large number of relations, the number of parameters can be huge. To avoid overfitting, (Schlichtkrull et al., 2018) introduces regularization techniques, which we omit due to the space constraint.

2.2 KG-BERT

KG-BERT (Yao et al., 2019) is an extension of BERT (Devlin et al., 2018) for knowledge graph completion tasks using the representations from the pre-trained language model where the triplets of knowledge graphs are treated as textual sentences. The tasks addressed by this paper are link prediction and relation prediction. For the link prediction task, the input of the KG-BERT is the tokens of the entity and relations of a triplet (s, r, o) and the task is to predict the true label of the triplet. For this task, the output of the KG-BERT model (s, r, o) is $C \in \mathbb{R}^H$ and it is passed through a classification layer that acts as a scoring function $f(s, r, o) =$ sigmoid (CW^T) where $W \in \mathbb{R}^{2\times H}$. Given a set of positive triplet $\mathbb{D}^+$ and a set of positive triplet $\mathbb{D}^-$ constructed accordingly, the loss function for the link prediction task is binary cross-entropy loss. For triplet $\tau = (s, r, o)$, ground-truth label y_τ and classification output $s_\tau \in \mathbb{R}^2$ where $s_{\tau 0} + s_{\tau 1} = 1$

$$\mathcal{L} = -\sum_{\tau\in\mathbb{D}^+\cup\mathbb{D}^-}(y_\tau\log(s_{\tau o}) + (1-y_\tau)\log(s_{\tau 1}))$$

and the negative samples $\mathbb{D}^-$ are generated as the following.

$$\begin{aligned}\mathbb{D}^- =& \{(h', r, t)|h' \in \mathbb{E} \wedge h' \neq h' \wedge (h', r, t) \notin \mathbb{D}^+\}\\ &\cup \{(h, r, t')|t' \in \mathbb{E} \wedge t' \neq t \wedge (h, r, t') \notin \mathbb{D}^+\}\end{aligned}$$

For the second task which is relation prediction, the input of the KG-BERT is the text representation of the entity s and t of a triplet (s, r, o) and the task is to predict the relation r where the classification layer is same as the link prediction one except the fact that relation prediction task is not binary and it's a multi-task classification hence $W \in R^{H\times C}$ and the loss function is a cross-entropy loss for multi-class. For a triplet $\tau = (s, r, o)$ with $y'_{\tau i} = 1$ when $r = i$ else $y'_{\tau i} = 0$ when $r \neq i$ and $s'_{\tau i}$ is the probability of triplet τ being in relation i.

$$\mathcal{L}' = -\sum_{\tau\in\mathbb{D}^+}\sum_{i=1}^{R} y'_{\tau i}\log(s'_{\tau i}) \qquad (1)$$

2.3 BERT-RGCN

For the R-GCN model, random embeddings are used as the initial embedding for the entities and relations and perform graph convolution on the knowledge graph to obtain the final representation for link prediction and relation prediction tasks. For the KG-BERT model, the representation obtained from the pre-trained language model is utilized for the link prediction and relation prediction task. Our proposal is to utilize the embeddings obtained from the BERT and KG-BERT and feed them to the R-GCN model and check how the change of the initialization changes the performance of the R-GCN. Specifically, for textual information of entity or relation, we obtain representations of all tokens in it and use the average as the representation of the entity or relation. Because of a memory constraint, we do not fine-tune the pre-trained embeddings but make them fixed during training and test. Following (Schlichtkrull et al., 2018), we employ DistMult (Yang et al., 2014) as the scoring function for link prediction tasks after obtaining the representation from R-GCN.

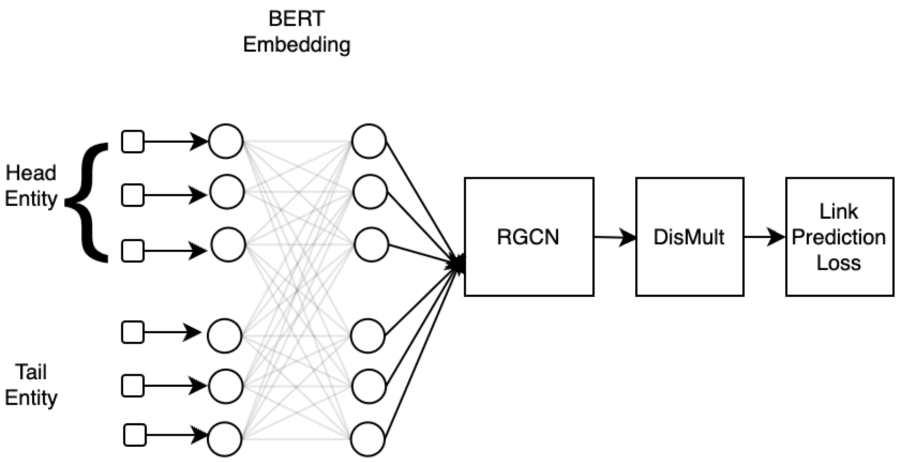


Figure 1: Architecture of our model BERT-RGCN for relation prediction from the entities. The entity embedding are obtained from the BERT embedding of the corresponding name or texts. After that, they are passed through a one-layer RGCN to perform graph convolution, and DisMult module is used to reduce the number of parameters and link prediction loss is used to update the parameters of RGCN and BERT

3 Experiments

We perform experiments on link prediction tasks to evaluate the prediction performance of our method on unknown relations. Following a standard procedure (Bordes et al., 2011, 2013; Schlichtkrull et al., 2018), for each triplet (s, r, o) in a test set, we generate negative examples by replacing either subject or object with incorrect entities, compute the prediction scores of true triplet and negative examples and construct a ranking, based on which ranking metrics are calculated over the test set.

3.1 Dataset

We use the following three datasets for our experiments. For each dataset, we randomly split the whole triplets into a training and test set, train a model on the training set and perform the evaluation on the test set.

WN18, WN18RR: WordNet (Miller, 1995) is a lexical database where semantic relations between a large number of English words are stored. WN18 is created by extracting relation of a subset of words from WordNet. WN18RR is a modified version of WN18 where triplets representing inverse relation to some triplets in the training set are removed from the test set to avoid leakage during testing. We use the definition of a word described in WordNet as an input to BERT. There are a few words that have long definitions, so we use the first 100 tokens for› each word after tokenizing a text with a pre-trained BERT model. For instance, the definition of an entity *toy* is *an artifact designed to be played with* and there is another entity *swing*, whose definition is *mechanical device used as a plaything to support someone swinging back and forth*. Also, their relation is given as *hyponym*. Both definitions contain some sense of playing, which we expect to help link prediction.

FB15k-237: Freebase has relationships between real-world entities such as people and place, from which the triplets of FB15k-237 dataset are extracted. Triplets representing inverse relation to some triplets in the training set are removed from the test set to avoid leakage during testing (Toutanova and Chen, 2015). Similar to (Yao et al., 2019), we use entity names as textual input to BERT. Examples of entity names are *United States of America* and *St. Augustine*, and their relation is labeled as /location/location/contains. As opposed to entity definitions in WN18 and WN18RR that consist of around 10 words and are descriptive, entity names are short and often proper nouns, which may make it more difficult for BERT-RGCN to obtain useful representation from textual information.

Properties	WN18	WN18RR	FB15k-237
Entities	40,943	40,943	14,541
Relations	18	11	237
Train edges	141,442	86,835	272,115
Val. edges	5,000	3,034	17,535
Test edges	5,000	3,134	20,466

Table 1: Dataset description.

Metrics	R-GCN	BERT-RGCN
MRR	**0.003353**	0.002122
Hit@1	**0.000729**	0.000298
Hit@3	**0.001943**	0.001043
Hit@10	0.006412	**0.006553**

(a) MRR and Hit@k for WN18 dataset. Training is done for 10000 epochs.

Metrics	R-GCN	BERT-RGCN
MRR	0.000228	**0.000693**
Hit@1	0.000002	**0.000160**
Hit@3	0.000005	**0.001043**
Hit@10	0.000008	**0.000957**

(b) MRR and Hit@k for WN18RR dataset. Training is done for 100 epochs.

Metrics	R-GCN	BERT-RGCN
MRR	**0.000798**	0.000657
Hit@1	**0.000400**	0.000024
Hit@3	**0.000600**	0.000513
Hit@10	**0.001200**	0.000684

(c) MRR and Hit@k for FB15k-237 dataset. Training is done for 10000 epochs. Due to a memory constraint, only 100 relations out of 237 are used.

Table 2: Performance Comparison.

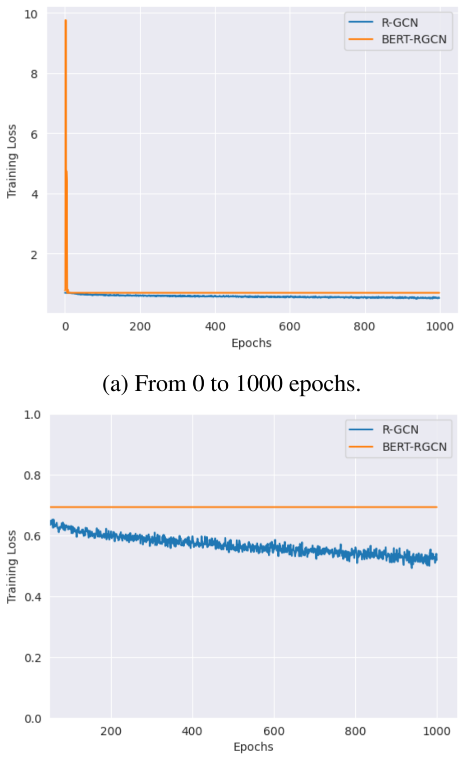
3.2 Training

We use BERT-tiny model (Bhargava et al., 2021; Turc et al., 2019) for initial entity and relation embeddings. We process a (sub-)graph with one graph-convolutional layer with 128 dimensions followed by ReLU activation to encode the entities and relations and apply DistMult (Yang et al., 2014) to get the score of each triplet. We perform training for 10000 epochs (100 epochs for WN18RR dataset) with mini-batches each of which consists of randomly sampled 1024 triplets.

3.3 Evaluation

We evaluate our model based on two metrics. One is *hit@n*, which is the fraction of entities in the test set scored in the top n in the rankings constructed with negative examples. The other metric is *mean reciprocal rank* (MRR). This is calculated by averaging the multiplicative inverse of the rank of the correct entity over the entities in the test set.

We compare the performance of our model with R-GCN. We run the experiments for R-GCN as well to make a comparison in the same experimental setting as BERT-RGCN, so the results provided in the following are not identical to the past results.



(a) From 0 to 1000 epochs.

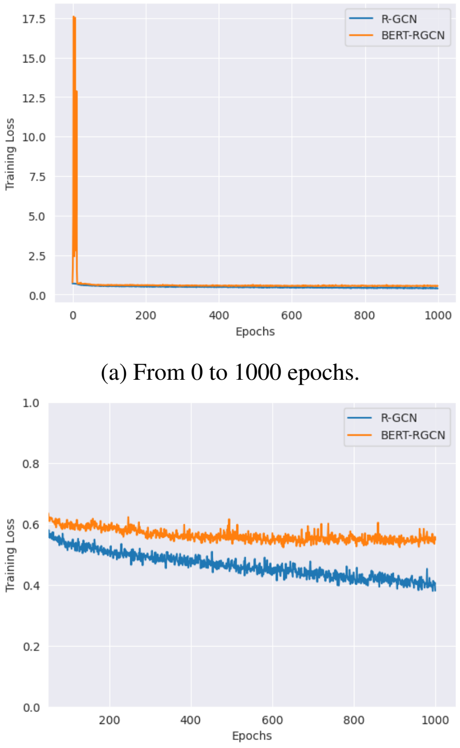
(b) From 50 to 1000 epochs.

Figure 2: Training Loss Curve for FB15k-237 dataset. Due to a memory constraint, only 100 relations out of 237 are used. (b) is provided because the loss of BERT-RGCN fluctuates in the first few epochs and it is hard to compare the latter epochs in (a).

3.4 Results

Table 2 shows the performance comprison for WN18, WN18RR and FB15k-237 datasets. From the results, we can observe that in our setting BERT-RGCN performs better than RGCN in the WN18RR dataset in the MRR and Hits@k metrics but for FB15k-237 and WN18 datasets, R-GCN shows better performance than our combined setting BERT-RGCN. Although textual information in FB15k-237 can be less useful than that of WN18 and WN18RR as we mentioned in 3.1, this is not sufficient to account for these results.

Figure 3 and 2 show the training loss over epochs for WN18 dataset and FB15k-237. Due to a memory constraint, only 100 relations and 15 relations are used for WN18 and FB15k-237 datasets respectively. From the figures, we can observe that the training loss of BERT-RGCN does not decrease after a few epochs, which can be considered to be a direct cause for the poor performance of BERT-RGCN. One possible reason for this is that since we do not fine-tune the pre-trained embeddings of



(a) From 0 to 1000 epochs.

(b) From 50 to 1000 epochs.

Figure 3: Training Loss Curve for WN18 dataset. Due to a memory constraint, only 15 relations out of 18 are used. (b) is provided because the loss of BERT-RGCN fluctuates in the first few epochs and it is hard to compare the latter epochs in (a).

BERT because of a memory constraint, the number of learnable parameters of BERT-RGCN is much smaller than that of R-GCN, resulting in severe underfitting for BERT-RGCN.

4 Discussion

Based on the idea that knowledge graph completion task should be improved by using both graph structure and textual information of entities, we combined pre-trained BERT embeddings with R-GCN and investigate its performance on link prediction tasks. Our experiments show that for one of the three datasets, BERT-RGCN performs better than R-GCN. However, for the other two datasets, the performance of BERT-RGCN is worse than that of R-GCN and our analysis implies that the model does not learn effectively during the training and poorly performs on the test set. We conjecture one reason for this is that we do not fine-tune the pre-trained embeddings due to a memory constraint, which may critically restrict the capacity of the model. For future work, we consider carefully studying the impact of fine-tuning with a smaller dataset and larger memory.

References

Prajjwal Bhargava, Aleksandr Drozd, and Anna Rogers. 2021. Generalization in nli: Ways (not) to go beyond simple heuristics.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26.

Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. 2011. Learning structured embeddings of knowledge bases. In *Proceedings of the AAAI conference on artificial intelligence*, volume 25, pages 301–306.

Jeffrey Dalton, Laura Dietz, and James Allan. 2014. Entity query feature expansion using knowledge base links. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 365–374.

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

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

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

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 *The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15*, pages 593–607. Springer.

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.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: The impact of student initialization on knowledge distillation. *CoRR*, abs/1908.08962.

Liang Wang, Wei Zhao, Zhuoyu Wei, and Jingming Liu. 2022. Simkgc: Simple contrastive knowledge graph completion with pre-trained language models. *arXiv preprint arXiv:2203.02167*.

Zihan Wang, Zhaochun Ren, Chunyu He, Peng Zhang, and Yue Hu. 2019. Robust embedding with multi-level structures for link prediction. In *IJCAI*, pages 5240–5246.

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

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

Xuchen Yao and Benjamin Van Durme. 2014. Information extraction over structured data: Question answering with freebase. In *Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 956–966.