Data Augmented Knowledge Graph Completion via Pre-trained Language Models

Shixuan Liu[1], Pouya Ghiasnezhad Omran[2] and Kerry Taylor[2]

[1]China Telecom Corporation Ltd. Data&AI Technology Company, 1301, 13 / F, Building 1, No. 31 Financial Street, Xicheng District, Beijing, China

[2]Australian National University, ACT 2601, Canberra, Australia

Abstract

Knowledge graphs provide significant assistance for many artificial intelligence tasks, but they are usually incomplete. Techniques for *knowledge graph completion* can improve the coverage of Knowledge Graphs (KGs) by inducing new facts. Traditional methods for completion use structural representations in embedding space, but textual information can also be helpful. Recently, pre-trained language models have shown impressive performance on natural language processing tasks. KG-BERT is a pre-trained language model that is used for knowledge graph completion and achieves appealing performance. However, KG-BERT struggles when the number of facts is inadequate. We consider the inadequacy of data for various relations and the compensation for sparsity via data augmentation. We propose two knowledge graph data augmentation methods to generate facts with novel relations. Specifically, multi-hop relations between two entities are extracted to form *multi-hop facts*, and *implicit relations* are generated by horn rules. Moreover, we find multi-hop facts are useful for few-shot learning scenarios. Our system improves the performance of KG-BERT regarding the accuracy of the link prediction task. The experimental results demonstrate that our models make significant enhancements for KG-BERT on several knowledge graph completion benchmarks (e.g., WN18RR and UMLS).

Keywords

Knowledge Graph Completion, Pre-trained Language Models, Data Augmentation

1. Introduction

Knowledge graphs (KG) are effective structures for machine learning, composed of ’triples’—pairs of entities and a relation. Given KGs’ inherent incompleteness, efforts are made to complete them by inferring plausible triples. Transformer-based models like BERT [1] have excelled in transferring pre-training knowledge to specific tasks. KG-BERT [1] is a notable development in this field, which employs pre-trained language models for KG Completion (KGC). It fine-tunes BERT models with concatenated text descriptions of triples and utilizes contextualized embedding to evaluate the validity of a potential fact.

Although KG-BERT demonstrates great achievement on the link prediction task, it can be further improved. Due to the incompleteness of knowledge graphs, some potential relationships could be further explored. For example, there are two triples *(Anthony Albanese, born_in_city,*

ISWC 2023 Posters and Demos: 22nd International Semantic Web Conference, November 6–10, 2023, Athens, Greece

liusx14@chinatelecom.cn (S. Liu); p.g.omran@anu.edu.au (P. G. Omran); kerry.taylor@anu.edu.au (K. Taylor)

0000-0002-4473-3877 (P. G. Omran); 0000-0003-2447-1088 (K. Taylor)

CEUR Workshop Proceedings (CEUR-WS.org)

Sydney) and *(Sydney, located_in, Australia)*. According to these two triples, *Anthony Albanese* and *Australia* can use an alternative relation that may not appear in the knowledge graph to directly represent their relationship, i.e., *nationality_of*. Besides, the potential relations explored by rules could also give further information. For example, consider the rule *co_work(X, Y)* $\leftarrow$ *work_at(X, A)* $\wedge$ *work_at(Y, A)*, where *X, Y, A* can be any entity that satisfies this rule and *co_work* and *work_at* are existing relations. Not all entities satisfying the rule's body will satisfy its head, it usually has a confidence score to show its reliability. If we want to find a relation that could describe *(X, Y)* while it should be semantically similar with *co_work*, like one relation expressing *work in the same company*. This kind of relation is a loose implicit expression of the original one, and it is worthwhile to investigate how to generate relations like it. While the rule learners can mine rules with relatively low confidence, we cannot apply all these rules to KG directly and treat all infrared facts as valid facts to KG.

In this paper, we improve the Knowledge Graph-BERT (KG-BERT) with two innovative data augmentation techniques. First, a multi-hop data augmentation approach boosts the overall performance and offers an effective solution for few-shot learning. Second, the creation of implicit facts using AnyBURL-generated horn rules. Such facts express implicit information by constructing similar relations by dropout. These strategies together significantly improve the robustness and performance of KG-BERT.

2. Related work

KG-BERT: BERT [2], as a landmark pre-trained language model, has achieved great performance on multiple NLP tasks. KG-BERT is an application of pre-trained language models on the knowledge graph completion task [1]. KG-BERT makes some changes to the fine-tuning process to suit the knowledge graph completion task. For a triple (h,r,t)(h, r, t), we could concatenate the head entity, relation, and tail entity as a sequence of tokens and encode them into word embeddings which can be represented as $\mathbf{W}^h = (\mathbf{w}^h_1, \mathbf{w}^h_2, ..., \mathbf{w}^h_{l_h})$, $\mathbf{W}^r = (\mathbf{w}^r_1, \mathbf{w}^r_2, ..., \mathbf{w}^r_{l_r})$ and $\mathbf{W}^t = (\mathbf{w}^t_1, \mathbf{w}^t_2, ..., \mathbf{w}^t_{l_t})$ respectively, where l_h, l_r and l_t are the length of head entity, relation and tail entity. Eventually, we concatenate sequences by special tokens **[CLS]** and **[SEP]** as $\widetilde{\mathbf{W}} = [\mathbf{w}^{[\mathrm{CLS}]}, \mathbf{W}^h, \mathbf{w}^{[\mathrm{SEP}]}, \mathbf{W}^r, \mathbf{w}^{[\mathrm{SEP}]}, \mathbf{W}^t, \mathbf{w}^{[\mathrm{SEP}]}]$. Meanwhile, different segment ids (0 and 1) are used to distinguish entity and relation as $[\mathbf{w}^{[\mathrm{CLS}]}(0), \mathbf{W}^h(0), \mathbf{w}^{[\mathrm{SEP}]}(0), \mathbf{W}^r(1), \mathbf{w}^{[\mathrm{SEP}]}(1), \mathbf{W}^t(0), \mathbf{w}^{[\mathrm{SEP}]}(0)]$. The head and tail entities have the same segment embeddings (type 0), which are different from the segment embeddings of the relation (type 1). The pooling output of **[CLS]** in the last hidden state will be seen as the representation of the whole triple and will be fed into a binary classifier to determine if the triple is real. In prediction, the positive probability of triples is used for ranking.

Rule Learning: Rule-based models, such as AnyBURL [3], can not only complete the knowledge graph but can also enable mining the rules, which may themselves be rich in information [4]. AnyBURL is based on sampling paths and generalizing them to Horn rules. For each generated ground path rule, its format is like $h(c_0, c_1) \leftarrow b_1(c_1, c_2), ..., b_n(c_n, c_{n+1})$, where $\{h, b_1, ...b_n\} \in \mathcal{R}$ and $\{c_0, c_1, ...c_n\} \in \mathcal{E}$. The head of the rule is $h(...)$ while body is $b_1(...)$ to $b_n(...)$. There are two types of generated rules, cyclic ($c_0 = c_{n+1}$) and acyclic ($c_0 \neq c_{n+1}$). The quality of early sampling can be improved by reinforcement learning. Meanwhile, the summarized

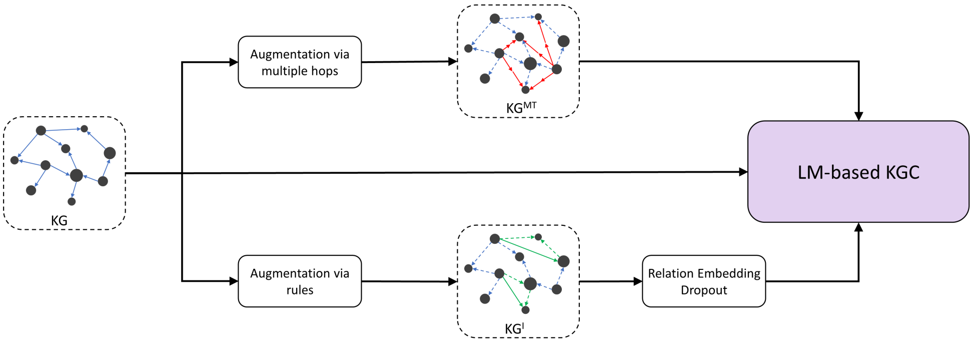

Figure 1: Flowchart showing how augmentation data complements language model-based knowledge graph completion

rules and corresponding confidence scores are applied for knowledge graph completion.

3. Augmentation for Language Model-based KGC

The flow chart shown in figure 1 shows how original data and two kinds of augmented data are fed into language models. For an LM-based KGC such as KG-BERT, we directly input the original knowledge graph KG into the model. For data augmentation via multi-hops, the multi-hop triples are extracted as a knowledge graph KG^{MT} and fed into the model. For data augmentation via rules, the implicit triples are extracted as a knowledge graph KG^{I}. Differently to KG and KG^{MT}, it first converts to word embeddings and is implemented dropout and then fed into the model.

Augmentation via Multi-hop Facts: For entities h and t, assume a series of n triples such as $\{(h, r_1, e_1)$, (e_1, r_2, e_2), ..., $(e_{n-1}, r_n, t)\}$. We can present their relation in a multi-hop format like $h\xrightarrow{r_1}e_1\xrightarrow{r_2} e_2\xrightarrow{...}e_{n-1}\xrightarrow{r_n}t$. In order to make multi-hop triples adaptable for pre-trained language models, we ignore the entities in the relation and get the triple $(h, r_1 \oplus r_2 \oplus ... \oplus r_n, t)$. We convert such triples suitable format and call this method **D**ata **A**ugmentation via **M**ulti-**H**op (DAMH). Assume a multi-hop triple has k relations as $(h, r_1 \oplus ... \oplus r_k, t)$, then the text sequence $\widetilde{\mathbf{W}}_{\text{DAMH}}$ can be represented as shown in formula 1. After that, we input $\widetilde{\mathbf{W}}_{\text{DAMH}}$ into a pre-trained language model and take out the pooled embedding $\mathbf{C}_{\text{DAMH}}$ as the representation of the entire multi-hop triple. Besides, constructing 2-hop triples for relations that infrequently appear in training dataset can effectively improve the performance, which we call this few-shot learning in knowledge graph completion [5].

$$\widetilde{\mathbf{W}}_{\text{DAMH}} = [\mathbf{w}^{[\text{CLS}]}, \mathbf{W}^{h}, \mathbf{w}^{[\text{SEP}]}, \mathbf{W}^{r_1}, \mathbf{w}^{[\text{SEP}]}, ..., \mathbf{W}^{r_k}, \mathbf{w}^{[\text{SEP}]}, \mathbf{W}^{t}, \mathbf{w}^{[\text{SEP}]}] \tag{1}$$

Augmentation via Horn Rules: Rule-based models such as AnyBURL can find potential triples by exploring horn rules. The information implied in these rules can also be used to generate augmented triples. By exploring implicit patterns in a knowledge graph, the rule-based model can obtain rules such as *co-work(X,Y)* ← *work_at(X,A)* ∧ *work_at(Y,A)*, where *co-work(X,Y)* is head and *work_at(X,A)* ∧ *work_at(Y,A)* is body. But not all entities X,Y,AX, Y, A satisfying body can get the head. If we need to find an implicit relation which can describe these entity pairs, then it should have some similarity to the original relation.

Inspired by SimCSE [6], there is a simple and effective way to construct implicit relations by dropout. We directly implement *relation embeddings dropout* for creating implicit relations, referred to **D**ata **A**ugmentation via **I**mplicit (DAI). As shown in formula 2 and 3, the dropout token embeddings added segment and position embeddings forms new token embeddings. We find even for graphs with only low-confidence rules, this approach can make improvements.

$$\widetilde{\mathbf{W}}_{\text{token}} = [\mathbf{w}_{\text{token}}^{[\text{CLS}]}, \mathbf{W}_{\text{token}}^{h}, \mathbf{w}_{\text{token}}^{[\text{SEP}]}, \text{Dropout}(\mathbf{W}_{\text{token}}^{r}, \mathbf{w}_{\text{token}}^{[\text{SEP}]}), \mathbf{W}_{\text{token}}^{t}, \mathbf{w}_{\text{token}}^{[\text{SEP}]}] \tag{2}$$

$$\widetilde{\mathbf{W}}_{\text{DAI}} = \widetilde{\mathbf{W}}_{\text{token}} + \widetilde{\mathbf{W}}_{\text{segment}} + \widetilde{\mathbf{W}}_{\text{position}} \tag{3}$$

4. Experiments and Conclusion

We conduct a set of experiments to assess the performance of our data augmentation methods. We evaluate our ***D**ata **A**ugmentation via **I**mplicit triples* method on four datasets, while two of them are open-source **WN18RR** [7] and **UMLS**[7]. Two of them are reconstructed for few-shot learning. **FB15k-237(few-shot)** takes part in the test triples in FB15k-237 [8], whose relations are top 50 infrequently appearing in the training dataset. **NELL-ONE(few-shot)** [9] are reconstructed as follows, for each set of triples in one relation, we separate in proportion (0.7, 0.1, 0.2) for training, validation, and test, respectively.

We have released our code and detailed results [1], and generated augmented datasets for training, favoring 2-hop data extraction for optimal performance. DAMH expands the training datasets of WN18RR and UMLS by 158% and 307% respectively. Statistics of datasets are also published in the link. We adopted KG-BERT's hyper-parameter settings due to high computational costs. Our methods are compared with other baselines on UMLS and WN18RR datasets. Test samples involved creating negative samples by replacing the head or tail entity with other entities not present in the training set.

Table 1
The experimental results of few-shot learning on FB15k-237 and reformatted NELL-ONE datasets

	WN18RR		**UMLS**		**FB15k-237 (few-shot)**		**NELL-ONE (few-shot)**	
Models	Hits@10	MR	Hits@10	MR	Hits@10	MR	Hits@10	MR
KG-BERT	52.4	97	99.0	1.47	50.7	142	17.4	**218**
KG-BERT$_{+DAI}$	56.8	90	99.3	1.46	-	-	-	-
KG-BERT+$_{+DAMH}$	62.4	**85**	99.5	**1.39**	**57.8**	**139**	**21.9**	258
KG-BERT+$_{+DAMH+DAI}$	**63.0**	156	**99.5**	1.42	-	-	-	-

From the comparison of experimental results in table 1, it can be seen that KG-BERT+DAMH, KG-BERT+DAI, and KG-BERT+DAMH+DAI all outperform the original KG-BERT on both

[1] Detailed experimental results and code can be found at https://github.com/LSX-Sneakerprogrammer/KG-Augmentation

WN18RR and UMLS datasets. For the Hits@10 metric, the most significant improvement on WN18RR is from KG-BERT+DAI+DAMH (+10.6% compared with KG-BERT), while for UMLS is from KG-BERT+DAMH (+0.5% compared with KG-BERT). For the mean rank metric, KG-BERT+DAMH outperforms the official result of KG-BERT by a significant 12 on WN18RR and a smaller 0.08 on UMLS. These results establish that our augmentation approaches make significant progress on these well-studied tasks.

For few-shot learning, we evaluate models on two datasets, FB15k-237 (few-shot) and NELL-ONE (few-shot), and compare them to KG-BERT. We can see the results in table 1 for FB15k-237 (few-shot). The improvements of KG-BERT+DAMH are 7.8% and 3, respectively for Hits@10 and Mean Rank. For the NELL-ONE (few-shot) dataset, the KG-BERT+DAMH surpasses KG-BERT on four metrics. These experimental results demonstrate the effectiveness of our method on few-shot learning.

Our study proposes generating augmented triples via multi-hop connections and horn rules, effectively enhancing KG-BERT, particularly with the multi-hop method. Both methods and their combined form remarkably improve performance, with the multi-hop approach notably strengthening few-shot learning tasks. Future research should investigate integrating the augmented data with other knowledge graph completion methods.

References

[1] L. Yao, C. Mao, Y. Luo, Kg-bert: Bert for knowledge graph completion, arXiv preprint arXiv:1909.03193 (2020).

[2] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: NAACL-HLT, 2019, pp. 4171–4186.

[3] C. Meilicke, M. W. Chekol, M. Fink, H. Stuckenschmidt, Reinforced anytime bottom up rule learning for knowledge graph completion, arXiv preprint arXiv:2004.04412 (2020).

[4] P. G. Omran, Z. Wang, K. Wang, Knowledge graph rule mining via transfer learning, in: Advances in Knowledge Discovery and Data Mining: 23rd Pacific-Asia Conference, PAKDD 2019, Macau, China, April 14-17, 2019, Proceedings, Part III 23, Springer, 2019, pp. 489–500.

[5] C. Zhang, H. Yao, C. Huang, M. Jiang, Z. Li, N. V. Chawla, Few-shot knowledge graph completion, in: Proceedings of the AAAI conference on artificial intelligence, volume 34, 2020, pp. 3041–3048.

[6] T. Gao, X. Yao, D. Chen, Simcse: Simple contrastive learning of sentence embeddings, in: EMNLP, 2021, pp. 6894–6910.

[7] T. Dettmers, P. Minervini, P. Stenetorp, S. Riedel, Convolutional 2d knowledge graph embeddings, in: AAAI, 2018, pp. 1811–1818.

[8] K. Toutanova, D. Chen, P. Pantel, H. Poon, P. Choudhury, M. Gamon, Representing text for joint embedding of text and knowledge bases, in: EMNLP, 2015, pp. 1499–1509.

[9] W. Xiong, M. Yu, S. Chang, X. Guo, W. Y. Wang, One-shot relational learning for knowledge graphs, in: EMNLP, 2018, pp. 1980–1990.