Multi-Task Learning for Knowledge Graph Completion with Pre-trained Language Models

Bosung Kim[1], **Taesuk Hong**[2], **Youngjoong Ko**[1], **Jungyun Seo**[2]
[1]Sungkyunkwan University, Suwon, Gyeonggi-do, Korea
[2]Sogang University, Seoul, Korea
{bosungkim17,lino.taesuk}@gmail.com
yjko@skku.edu, seojy@sogang.ac.kr

Abstract

As research on utilizing human knowledge in natural language processing has attracted considerable attention in recent years, knowledge graph (KG) completion has come into the spotlight. Recently, a new knowledge graph completion method using a pre-trained language model, such as KG-BERT, was presented and showed high performance. However, its scores in ranking metrics such as Hits@k are still behind state-of-the-art models. We claim that there are two main reasons: 1) failure in sufficiently learning relational information in knowledge graphs, and 2) difficulty in picking out the correct answer from lexically similar candidates. In this paper, we propose an effective multi-task learning method to overcome the limitations of previous works. By combining relation prediction and relevance ranking tasks with our target link prediction, the proposed model can learn more relational properties in KGs and properly perform even when lexical similarity occurs. Experimental results show that we not only largely improve the ranking performances compared to KG-BERT but also achieve the state-of-the-art performances in Mean Rank and Hits@10 on the WN18RR dataset.

1 Introduction

A Knowledge Graph (KG) is a graph-structured knowledge base, where real-world knowledge is represented in the form of triple (h, r, t): *(head entity, relation, tail entity)* which means h and t have a relationship r. Entities and the relation in a triple are denoted as nodes and an edge of the graph, respectively. In recent years, Natural Language Processing (NLP) has benefited from utilizing KGs in various applications such as language modeling (Peters et al., 2019; Liu et al., 2019a), question answering (Zhang et al., 2019; Huang et al., 2019), and machine reading (Yang and Mitchell, 2017). Since there has been an increasing demand for high-quality knowledge, the reliability of KG has also become important. Therefore, knowledge graph completion (a.k.a. link prediction), which identifies whether the triple in KG is valid or not, has been actively investigated.

Several studies on the knowledge graph completion have been conducted (Bordes et al., 2013; Trouillon et al., 2016; Sun et al., 2019; Dettmers et al., 2018). They presented methods to model the connectivity patterns between entities in KG, and score functions to define the validity of the triple. However, these methods only consider graph structure and relational information depending on existing KG. Thus, they cannot predict well on triples that contain less frequent entities. Recently, addressing the sparseness problem of previous models, Yao et al. (2019) proposed a method called KG-BERT for knowledge graph completion, using entity descriptions and pre-trained language models. Even though KG-BERT significantly improved mean ranks using preliminary linguistic information from BERT (Devlin et al., 2018), the results in other ranking metrics such as MRR and Hit@k are still behind the state-of-the-art models.

We claim that there are two major reasons for this problem. First, KG-BERT misses lots of relation information in KGs. While previous state-of-the-art methods aimed to model relational properties in graphs, KG-BERT only uses binary cross entropy loss to predict valid or invalid triples for the link prediction task. Next, KG-BERT has difficulty in picking out the answer entity between lexically similar

Proceedings of the 28th International Conference on Computational Linguistics, pages 1737–1743
Barcelona, Spain (Online), December 8-13, 2020

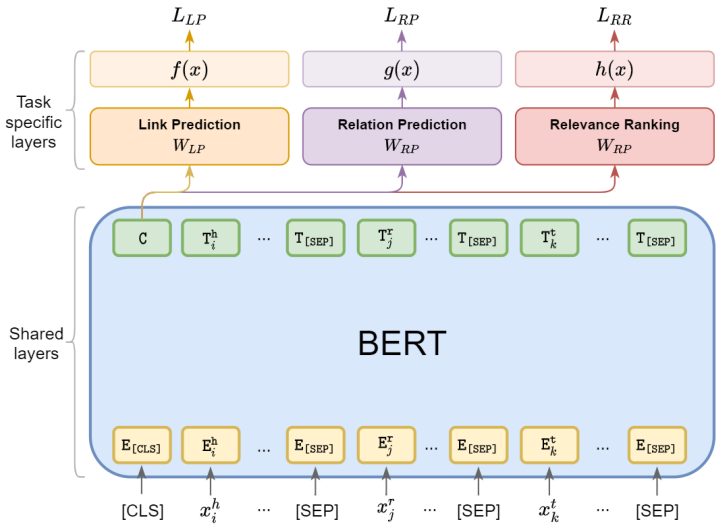

Figure 1: Architecture of the proposed multi-task learning method for knowledge graph completion.

candidates. For example, given head entity and relation as *(take a breather, derivationally related for, _)* and the correct tail entity as *"breathing time"*, KG-BERT predicts *"snorkel breather"* and *"breath"* as top scores because of the lexical similarity by *"breath"*. This problem leads to lower performance in MRR and Hits@k.

In this paper, we propose an effective multi-task learning method to overcome these problems. We devise a multi-task framework by adding two tasks (relation prediction and relevance ranking) to link prediction, our target task. In the relation prediction, the model is trained to predict the relationship between given two entities, which helps the model learn more relational properties. In the relevance ranking, the model is trained by the margin ranking loss to make a gap between the valid triple and lexically similar candidates. We evaluate the proposed method on two popular datasets WN18RR and FB15k-237, and experimental results show that our method could improve ranking performance by a large margin compared to KG-BERT. Notably, our method achieves state-of-the-art performances in Mean Rank and Hits@10 on the WN18RR dataset.

2 Proposed Method

In this section, we propose a multi-task learning for knowledge graph completion. As shown in Figure 1, we follow a multi-task learning framework in MT-DNN (Liu et al., 2019b), and use the pre-trained BERT model as a shared layer. We combine three tasks: link prediction, relation prediction, and relevance ranking. Each task has a classification layer $W \in \mathbb{R}^{K \times H}$ where K is the number of labels and H is the hidden size of BERT. Following Devlin et al. (2018), every input sequence has a `[CLS]` token at the head of sentence, and `[SEP]` token is used as a separator.

Link Prediction (LP): We define link prediction as same as KG-BERT (Yao et al., 2019), and this is our main target task. Given a training set S, the input x is a text sequence of (h, r, t). Each entity is represented as entity name and description, e.g., for triple *(plant tissue, hypernym, plant structure)*, the input sequence is as follows:

> `[CLS]` plant tissue, the tissue of a plant `[SEP]` hypernym `[SEP]` plant structure, any part of a plant or fungus `[SEP]`

The model is trained to predict whether a given triple (h, r, t) is valid or not, and invalid triples are made by replacing head or tail entity with one of random entities. Let C be the final hidden vector of `[CLS]` token, $W_{LP} \in \mathbb{R}^{2 \times H}$ be a classification layer for link prediction, and S' be a invalid triple set, then

$$f(x) = \text{softmax}(CW_{LP}^T) = [\widehat{y}_0, \widehat{y}_1], \qquad \mathcal{L}_{LP} = -\sum_{x \in \{S \cup S'\}} y \log \widehat{y}_1 + (1-y) \log \widehat{y}_0 \tag{1}$$

	# of entities	# of relations	train	validation	test
WN18RR	40,943	11	86,835	3,034	3,134
FB15k-237	14,541	237	272,115	17,535	20,466

Table 1: Statistics of datasets.

where $f(x)$ is the final output of the model and $y \in \{0, 1\}$ is a label. Let the output of CW_{LP}^T be $[s_0, s_1] \in \mathbb{R}^2$, then s_1 is used as the final ranking score in evaluation.

Relation Prediction (RP): The model learns to classify the relation of two entities. The input is head and tail entity sequences, e.g., "`[CLS]` plant tissue, the tissue of a plant `[SEP]` plant structure, any part of a plant or fungus `[SEP]`", then the model trains to predict the relation *hypernym*. The classification layer for relation prediction is $W_{RP} \in \mathbb{R}^{R \times H}$ where R is the number of relations, and we minimize a cross-entropy loss.

$$g(x) = \text{softmax}(CW_{RP}^T), \qquad \mathcal{L}_{RP} = -\sum_{x \in S} y \log g(x) \tag{2}$$

where $g(x)$ is the output of the model and $y \in \mathbb{R}^R$ is a class indicator.

Relevance Ranking (RR): The objective of relevance ranking is to make valid triples keep higher scores than invalid triples. We use a margin ranking loss to provide a bigger gap between valid and invalid triples. The input is the same as link prediction, and the classification layer for relevance ranking is $W_{RR} \in \mathbb{R}^{1 \times H}$.

$$h(x) = \text{sigmoid}(CW_{RR}^T), \qquad \mathcal{L}_{RR} = \sum_{x \in S, x' \in S'} \max\{0, h(x') - h(x) + \lambda\} \tag{3}$$

where $h(x)$ is the output of the model and λ is a margin.

In the training time, we use mini-batch based stochastic gradient descent. We first compose mini-batches for each task, D_{LP}, D_{RP}, and D_{RR}, then combine all data $D = D_{LP} \cup D_{RP} \cup D_{RR}$. At each training step, the mini-batch is randomly selected from D, and then the task corresponding to the batch is trained sequentially.

3 Experiments

Datasets We evaluated the proposed multi-task learning method on two benchmark datasets WN18RR (Dettmers et al., 2018) and FB15k-237 (Toutanova and Chen, 2015). Each dataset consists of a set of triples in the form of (h, r, t). WN18RR is a subset of WordNet, which is a lexical database of English. Thus, entities in WN18RR are words or short phrases, and there exists 11 relations between two words, such as *hypernym* and *similar to*. FB15k-237 is a subset of Freebase (Bollacker et al., 2008), a large-scale graph database including general human knowledge. FB15k-237 has more general entities, such as *Lincoln* and *Monaco*, and relations are longer and more complex than WN18RR. We used the same entity descriptions with Yao et al. (2019): synset definitions from WordNet for WN18RR and descriptions from Xie et al. (2016) for FB15k-237. Table 1 summarizes our datasets.

Baselines We mainly compare our method with KG-BERT (Yao et al., 2019), and also provide a comparison with several outstanding models: TransE (Bordes et al., 2013), DistMult (Yang et al., 2014), ComplEx (Trouillon et al., 2016), ConvE (Dettmers et al., 2018), and RotatE (Sun et al., 2019).

Experimental Settings We used pre-trained BERT-base as a shared layer and fine-tuned over the multi-task setup for 3 epochs. We used mini-batch size of 32 and Adam optimizer (Kingma and Ba, 2014) with learning rate 2e-5. In relevance ranking, we set the margin λ on the validation set, and it showed best results when $\lambda = 0.1$.

Evaluation Settings We evaluate our method on the link prediction, where the model predicts the head entity given $(\_, r, t)$ and tail entity given $(h, r, \_)$. To compare prior work, we follow the evaluation protocol and *filtered setting* in Bordes et al. (2013). Let $\mathbb{E}$ be a entity set and $\mathbb{T}$ be a set of all triples in train, valid, and test. Then, the set of test candidates U for predicting h in a given triple (h, r, t) is

	WN18RR					FB15k-237				
	MR	MRR	Hits@1	Hits@3	Hits@10	MR	MRR	Hits@1	Hits@3	Hits@10
KG-BERT (Yao et al., 2019)	97	-	-	-	52.4	153	-	-	-	42.0
KG-BERT (our results)	108	21.9	9.5	24.3	49.7	145	23.7	14.4	26.0	42.7
LP + RP	112	30.2	17.7	35.3	56.0	138	26.2	16.9	28.9	44.7
LP + RR	97	27.7	13.0	34.1	57.6	143	24.7	15.4	27.2	43.4
LP + RP + RR	**89**	**33.1**	**20.3**	**38.3**	**59.7**	**132**	**26.7**	**17.2**	**29.8**	**45.8**

Table 2: Link prediction results on WN18RR and FB15k-237. The second row shows our results of KG-BERT under the same implementation and hyperparameter settings as the original work.

KG-BERT			Ours		
rank	entity	score	rank	entity	score
1	snorkel breather	4.128	1	**breathing time**	4.541
2	breath	4.119	2	rest	4.442
3	artificial respiration	4.118	3	relaxer	4.359
4	respirator	4.114	4	time out	4.333
5	relaxer	4.106	5	respirator	4.271
6	take a breath	3.991	6	breath	4.195
	...			...	
22	**breathing time**	3.804			

Table 3: Example of results for the triple *(take a breather, derivationally related for, breathing time)* in WN18RR. Given *(take a breather, derivationally related for, _)*, the answer is **breathing time**.

	WN18RR		FB15k-237	
	MR	Hits@10	MR	Hits@10
TransE	2365	50.5	223	47.4
DistMult	3704	47.7	411	41.9
ComplEx	3921	48.3	508	43.4
ConvE	5277	48.0	246	49.1
RotatE	3340	57.1	177	**53.3**
KG-BERT	97	52.4	153	42.0
Ours	**89**	**59.7**	**132**	45.8

Table 4: Comparison with previous state-of-the-art models. Results are taken from Yao et al. (2019).

U = *(h, r, t)* $\cup \{(h', r, t)|h' \in \mathbb{E} \wedge (h', r, t) \notin \mathbb{T}\}$ and U for predicting t is U = *(h, r, t)* $\cup \{(h, r, t')|t' \in \mathbb{E} \wedge (h, r, t') \notin \mathbb{T}\}$. After the model computes scores of all candidate triples, they are sorted in descending order. The performances are evaluated in Mean Rank (MR), Mean Reciprocal Rank (MRR), and Hits@1, 3, 10.

3.1 Main Results

Table 2 demonstrates how the proposed method improves performance over the baseline model on the link prediction. The results show that multi-task learning with two tasks (LP + RP) and (LP + RR) could improve over the baseline by a large margin maintaining low MR scores. When the model is trained on three tasks (LP + RP + RR), we gain significant improvements, especially in Hits@1 and Hits@3 with 10.8 and 14.0, respectively. Table 3 shows an example of results in WN18RR. We observe that our model can choose the correct answer *"breathing time"* as the first ranking among lexically similar words, while the KG-BERT predicts *"snorkel breather"* and *"breath"* in top ranks. More examples are presented in Appendix A.

In the FB15k-237 benchmark, the task becomes more challenging as the number of relations increases up to 237, whereas the WN18RR contains only 11 relations. Thus, joint training with Relation Prediction (RP) was more effective on the FB15k-237, and this is shown as results that the model outperformed the baseline by 7, 2.5, 2.5, 2.9, and 2 absolute scores on MR, MRR, Hits@1, Hits@3, and Hits@10, respectively. When the Relevance Ranking (RR) task is added, and the model is trained with three different tasks, it achieves further improvements in all metrics with 13, 3, 2.8, 3.8, and 3.1 points, respectively.

A Comparison with previous models is presented in Table 4. Our model achieved state-of-the-art performances in MR and hits@10 on the WN18RR. In the FB15k-237 dataset, the performance of our model is lower than that of several models in Hits@10. Since FB15k-237 has more relations and a more complex graph structure than WN18RR, we conjecture that pre-trained language models cannot capture the complex structural information in knowledge graphs. Despite that, we achieved the best MR score on FB15k-237.

4 Related Work

A common approach for the knowledge graph completion is learning vector embeddings of the entities and the relationships in KG (Bordes et al., 2013; Yang et al., 2014; Trouillon et al., 2016; Sun et al., 2019; Dettmers et al., 2018). The most widely used method is TransE (Bordes et al., 2013), which models the relationships as translations in low-dimensional vector space. Dettmers et al. (2018) and Nguyen et al. (2018) proposed the embedding models using a convolutional neural network. Recent research has shown that the relation in complex vector space can infer the connectivity patterns: symmetry/antisymmetry, inversion, and composition (Sun et al., 2019). On the one hand, Yao et al. (2019) proposed KG-BERT that uses pre-trained language models (PLM) with entity descriptions. It can capture the contextualized meaning of entities and significantly improve mean ranks with rich linguistic information from PLM.

Multi-task learning has gained popularity over a decade in natural language processing (Collobert and Weston, 2008; Luong et al., 2015; Hashimoto et al., 2017; Liu et al., 2019b) of various tasks. It aims to regularize deep learning models from overfitting by sharing parameters of different tasks while jointly training them. With the advent of powerful PLMs such as BERT (Devlin et al., 2018) and XLNet (Yang et al., 2019), a multi-task learning scheme is applied by sharing pre-trained parameters of these models when training different tasks simultaneously.

5 Conclusion and Future Work

We propose an effective multi-task learning method for knowledge graph completion by combining relation prediction and relevance ranking tasks with link prediction. Experimental results demonstrate that our method outperforms previous strong baselines, and we largely improve MRR and Hits@k compared to the previous KG-BERT model.

In the future, we plan to investigate how to combine pre-trained language models and graph embedding methods to fully utilize the prior linguistic information of pre-trained models and graph structural information.

Acknowledgements

This work was supported by Institute for Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. 2020-0-00368, A Neural-Symbolic Model for Knowledge Acquisition and Inference Techniques).

References

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, page 1247–1250, New York, NY, USA. Association for Computing Machinery.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2787–2795. Curran Associates, Inc.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 160–167, New York, NY, USA. Association for Computing Machinery.

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

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*.

Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. 2017. A joint many-task model: Growing a neural network for multiple NLP tasks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1923–1933, Copenhagen, Denmark, September. Association for Computational Linguistics.

Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. 2019. Knowledge graph embedding based question answering. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, page 105–113, New York, NY, USA. Association for Computing Machinery.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2019a. K-bert: Enabling language representation with knowledge graph. *arXiv preprint arXiv:1909.07606*.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019b. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496, Florence, Italy, July. Association for Computational Linguistics.

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114*.

Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung. 2018. A novel embedding model for knowledge base completion based on convolutional neural network. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 327–333, New Orleans, Louisiana, June. Association for Computational Linguistics.

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

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*.

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

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

Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. 2016. Representation learning of knowledge graphs with entity descriptions. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Bishan Yang and Tom Mitchell. 2017. Leveraging knowledge bases in LSTMs for improving machine reading. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1436–1446, Vancouver, Canada, July. Association for Computational Linguistics.

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*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32*, pages 5753–5763. Curran Associates, Inc.

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

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

Appendix A Examples of the results in Link Prediction

E1. Given *(take a breather, derivationally related form, _)*, the answer is **breathing time**

TransE			RotatE			KG-BERT			Ours		
rank	entity	score	rank	entity	score	rank	entity	score	rank	entity	score
1	take a breather	5.9748	1	respiratory	0.0338	1	snorkel breather	4.1283	1	**breathing time**	4.5419
2	rest	4.0482	2	respirator	-0.2770	2	breath	4.1192	2	rest	4.4420
3	pause	1.2906	3	caesura	-0.4172	3	artificial respiration	4.1181	3	relaxer	4.3598
4	rest	0.3586	4	blackout	-0.4752	4	respirator	4.1147	4	time out	4.3331
5	respire	0.2834	5	intake	-0.6387	5	relaxer	4.1060	5	respirator	4.2710
6	rest	0.2108	6	time out	-0.7105	6	take a breath	3.9910	6	breath	4.1959
	...			...			...			...	
9	**breathing time**		11	**breathing time**		22	**breathing time**				

E2. Given *(_, hypernym, piece of music)*, the answer is **andante**

TransE			RotatE			KG-BERT			Ours		
rank	entity	score	rank	entity	score	rank	entity	score	rank	entity	score
1	piece of music	2.8572	1	piece of music	1.4983	1	composition	4.1524	1	sonata	3.9856
2	melodize	0.9326	2	**andante**	0.6013	2	piece of music	4.1427	2	piece of music	3.9836
3	soloist	0.8523	3	music	0.3708	3	finale	4.1155	3	**andante**	3.9605
4	realize	0.7959	4	tune	-0.0011	4	theme	4.1149	4	harmonization	3.9135
5	write	0.7664	5	serenade	-0.1049	5	**andante**	4.1021	5	composition	3.9127
6	score	0.6901	6	tucket	-0.1136	6	recapitulation	4.0845	6	finale	3.7482
7	**andante**	0.6199	7	strain	-0.1219	7	sonata	4.0736	7	fragment	3.7108

E3. Given *(systems software, hypernym, _)*, the answer is **programme**

TransE			RotatE			KG-BERT			Ours		
rank	entity	score	rank	entity	score	rank	entity	score	rank	entity	score
1	systems software	2.8572	1	systems software	-9.8255	1	systems software	4.1226	1	**programme**	4.1257
2	influence	-9.7353	2	location	-10.0784	2	**programme**	4.1201	2	utility program	4.0341
3	concert	-9.7715	3	horse	-10.1251	3	applications programme	4.1146	3	programme	4.0108
4	tap	-9.7720	4	learned profession	-10.1345	4	utility program	4.1045	4	software system	3.9995
5	fellow traveller	-9.7767	5	chemistry	-10.1458	5	compiling program	4.1024	5	programming	3.9647
6	landing	-9.7976	6	officiate	-10.1596	6	object-oriented programming language	4.0856	6	systems software	3.8621
	...			...							
8235	**programme**		16452	**programme**							

Table 5: Examples of results in Link Prediction.

For the example 1, the entity *breathing time* appears only once in the training set. Thus, the methods using only graph structure information, such as TransE and RotatE, cannot predict well on the given triple. Our model provides the correct answer, while KG-BERT predicts *snorkel breather* and *breath* as top scores due to the lexical similarity by *breath*. In example 2, the entity *piece of music* has lots of relationships with other entities; thus, most models show low performance on that example. Lastly, the example 3 shows that how the pre-trained language model (PLM) improves Mean Rank significantly. KG-BERT and our model give a high score for the answer *programme* using preliminary linguistic information from PLM, but the results of TransE and RotatE are extremely low.

Appendix B Computing Infrastructure

We ran all experiments on a single NVIDIA Titan RTX (24GB) with CUDA 10.1 version.

Appendix C Implementation

The source code of the paper is available at `https://github.com/bosung/MTL-KGC`.