Combining Structure Embedding and Text Semantics for Efficient Knowledge Graph Completion

Wen Sun
School of Informatics
Xiamen University
Xiamen, China
wensun.cs@gmail.com

Yifan Li
School of Film
Xiamen University
Xiamen, China
liyifan422@stu.xmu.edu.cn

Junfeng Yao
School of Film
Xiamen University
Xiamen, China
yao0010@xmu.edu.cn

Qingqiang Wu*
School of Informatics
School of Film
Xiamen University
Xiamen, China
wuqq@xmu.edu.cn

Kunhong Liu*
School of Film
Xiamen University
Xiamen, China
lkhqz@xmu.edu.cn

***Abstract*—Knowledge graph completion plays a crucial role in downstream applications. However, existing methods tend to only rely on the structure or textual information, resulting in suboptimal model performance. Moreover, recent attempts to leverage pre-trained language models to complete knowledge graphs have proved unsatisfactory. To overcome these limitations, we propose a novel model that combines structural embedding and semantic information of the knowledge graph. Compared with previous works based on pre-trained language models, our model can better use the implicit knowledge of pre-trained language models by using relation templates, entity definitions, and learnable tokens. Furthermore, our model employs a multi-head attention mechanism to transform the embedding semantic space of entities and relations obtained from the knowledge graph embedding model, thereby enhancing their expressiveness and unifying the semantic space of both types of information. Finally, we utilize convolutional neural networks to extract features from the matrices created by combining these two types of information for link prediction and triplet classification tasks. Empirical evaluations on two knowledge graph completion datasets demonstrate that our model is effective for both tasks.**

***Index Terms*—Knowledge graph completion, Knowledge graph, Link prediction**

I. Introduction

A knowledge graph (KG) is a structured representation of the objective world that captures information about objects and their relations, typically composed of fact triples that describe the relations between head entities and tail entities [1]. KGs have a significant impact on various natural language processing tasks [2], such as question answering, information retrieval, and recommendation systems. However, the challenge of the incompleteness problem in KGs has long impeded their effectiveness in various downstream applications.

To address this problem, researchers have turned to knowledge graph completion (KGC) methods, which aim to predict missing entities or relations in factual triples. Building on the success of word embeddings in capturing semantic information, knowledge graph embedding (KGE) models have been developed to address the link prediction problem [3], [4]. These methods treat entities and relations as continuous low-dimensional embeddings that can effectively preserve the semantics and intrinsic structure of entities and relations, allowing for computable representations. By fully comprehending the existing structures in the knowledge graph, the KGE models achieve missing link prediction by designing corresponding scoring functions and learning low-dimensional continuous vector representations of entities and relations. This methodology enables effective knowledge graph completion by accurately predicting missing entities or relations.

KGE has become the most popular method for KGC due to its simplicity and efficiency. However, this approach is limited to using structural information from existing KGs and is not effective for predicting entities and relations that are not present in the training set, thus making it less suitable for completing sparse knowledge graphs. Therefore, it is crucial to incorporate relevant textual information, such as entity and relation definitions and descriptions, to enrich the representation vector [5], [6].

Recent advancements in deep learning have led to the development of pre-trained language models (PLMs) such as BERT [7], which have shown outstanding performance in natural language processing tasks by learning word embeddings containing rich contextual semantic information from large-scale natural language text data. As a result, using PLMs to encode text data in knowledge graphs has attracted considerable attention. However, most PLM-based KGC models [5], [8] simply concatenate entity and relation labels as model inputs, failing to take full advantage of the implicit knowledge contained within PLMs and resulting in ineffective models.

Based on the aforementioned issues, we propose SS-KGC, a novel model for KGC that integrates both structural embedding and textual semantic information. Our model utilizes PLMs to encode textual data in triples, which are then transformed into coherent sentences using relation templates. Additionally, entity definitions are incorporated to better express the semantics of the triples, and learnable tokens [11] are added to improve

DOI reference number: 10.18293/SEKE2023-100

*Corresponding author

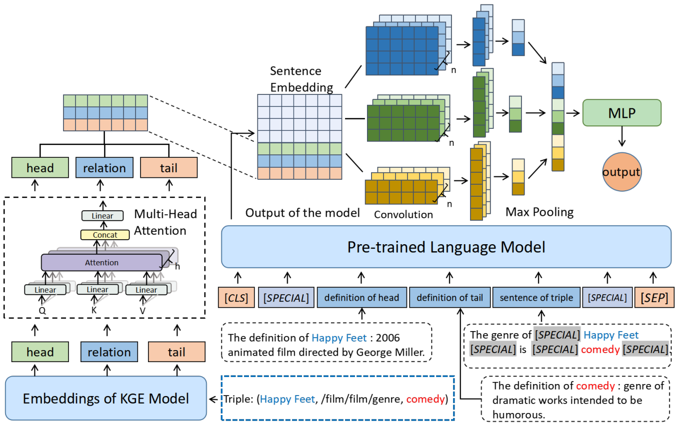

Fig. 1. The framework of SS-KGC.

the model's effectiveness.

Furthermore, the model employs a multi-headed attention mechanism [9] to transform the semantic space of the embedding vectors of entities and relations obtained through a basic KGE model. This enhances the model's expressive ability and unifies the space of semantic and structural information, making their integration more reasonable. Finally, the model utilizes convolutional neural networks (CNN) for feature extraction and is evaluated on link prediction and triadic classification tasks [10].

Experimental results demonstrate that our model outperforms the baseline models on two benchmark test sets for KGC. In summary, our contributions are as follows:

- We propose a novel KGC model based on PLMs, which combines the structural and semantic information for KGC and has achieved excellent results in experiments.
- We experimentally demonstrate the effectiveness of our model to combine semantic information and structural embedding of knowledge graphs for KGC.

II. Method

We propose SS-KGC to combine structural embedding and text semantic information for KGC. As shown in Figure 1, SS-KGC effectively utilizes both textual and structural information from the knowledge graph for KGC. Text semantic information comes from the labels and definitions in knowledge graphs. The structural embedding is obtained by pre-training the KGE model.

A. Data Processing

As our model utilizes BERT to encode text information, it is necessary to convert the triple data format of the knowledge graph into the input format of the BERT. For the triple data, it includes meaningful entity and relation labels and entity definitions, so it needs to deal with combining entity and relation labels into sentences and adding entity definitions into them. When converting entity and relation labels into sentences, we adopt the method of designing different conversion templates for different relations to convert triples into

TABLE I
Partial relations conversion template.

Relation	template
/film/film/genre	The genre of [X] is [Y] .
/music/genre/artists	The music genre of [Y] is [X] .
/people/person/gender	The gender of [X] is [Y] .
/film/film/country	The country [X] belongs to is [Y] .
/film/film/language	The language of the film [X] is [Y] .

coherent sentences [6], avoiding the problem of incoherent sentences caused by direct splicing, so that the model can better encode contextual semantic information. Table I shows the transformation templates of some relations, where $[X]$ represents the head entity and $[Y]$ represents the tail entity.

To fully express the semantic information of triples, we consider incorporating the definitions of entities into the model and introducing special tokens in the BERT vocabulary as learnable tokens [11]. These tokens are learnable during the training process, allowing the model to adapt to specific tasks and data by learning additional information that is not fixed. By using learnable tokens, the model can learn new representations suitable for the task when training data is limited or tasks are complex, which improves the model's ability to generalize to new data and make accurate predictions.

B. Text Semantic Encoding

The text data of the KG is integrated into coherent sentences after passing through the data processing layer as the input of the PLM. We use the BERT [7] as the encoding model. After the tokenizer of the BERT and the data processing operations, the input of the model is as follows:

$$S = [CLS], [SPECIAL], E_{definition}, T_{sentence}, [SPECIAL], [SEP] \quad (1)$$

Among them, the input sentence begins with the $[CLS]$ special token and ends with the $[SEP]$. $E_{definition}$ are the entity definitions, which are used to supplement the information of the head and tail entities. $[SPECIAL]$ is a special token added to the BERT vocabulary as a learnable token to make the model more effective. $T_{sentence}$ is the sentence transformed by the designed relational template. As shown in Figure 1, taking the triple (Happy Feet, /film/film/genre, comedy) as an example, the transformed sentence is "The genre of Happy Feet is comedy.".

Then, we use BERT to dynamically represent the word vector. Through its bidirectional encoding capability, the model can effectively obtain word embedding containing rich contextual semantic information. After multi-layer encoding, the sentence embedding matrix containing rich semantic information is finally obtained. Its dimension is $d_b \times d_l \times d_k$, which d_b is the size of the data batch, and d_l is the sentence length in BERT, d_k is the word embedding dimension of the BERT.

C. Knowledge Graph Embedding

The knowledge graph embedding layer mainly designs a simple KGE model to obtain the structure embedding of entities and relations, which is used as the structural information of the model for KGC. Based on the improvement of the TransE [3] model, we use $f = h \times r - t$ as the KGE score function.

To enrich the expressive ability of the embedding of entities and relations, after obtaining the embeddings of entities and relations, the semantic space is transformed through the multi-head attention mechanism [9], and the semantic space of the two types of information is unified by the linear layer of the last layer. The number of heads in our experiment is 3. The self-attention is calculated as follows:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d}})V \quad (2)$$

Where Q is the query vector, K is the key-value vector, V is the value vector, and d is the embedding dimension. When using the multi-head attention mechanism, multiple self-attention operations are performed on the same vector sequence. Each group uses a different parameter matrix to calculate attention separately and obtain multiple outputs. The calculation formula is as follows:

$$head_i = Attention(QW_{Qi}, KW_{ki}, VW_{Vi}) \quad (3)$$

Among them, W_{Qi}, W_{ki}, W_{Vi} are the ith mappings. These outputs are finally concatenated together.

$$Multihead = Concat(head_1, head_2, ..., head_h)W_o \quad (4)$$

Among them, the parameter matrix W_o is used to unify the semantic space output by the independent attention mechanism.

D. Convolutional Neural Network Layers

After obtaining both types of information, we utilize a convolutional neural network layer for further feature extraction in downstream tasks such as link prediction and triple classification [10]. The key advantage of CNNs is their ability to capture local features, which are automatically combined and filtered to obtain semantic information at different levels, as reflected in the text by N-gram features. Additionally, CNNs achieve good results with faster training speeds due to their parameter-sharing property.

The CNN layer takes as input the sentence embedding matrix obtained from combining the semantic and structural information of the knowledge graph. The input matrix is convolved using 256 convolution kernels of sizes 2, 3, and 4, and then pooled using global max pooling in the pooling layer. Finally, the global feature vector is mapped to the probability distribution of the output category using a fully connected layer for classification.

E. Training

The model performs negative sampling through a combination of two negative sampling strategies [6]: (1) Randomly replace the head or tail entity of the triple with other entities to generate a triple that does not exist in the knowledge graph. (2) Using the KGE model to replace the head or tail entity with other entities with high confidence produces triples that do not exist in the knowledge graph, thereby improving the quality of negative samples. Because the latter method is relatively complex, each of these two methods provides 50% negative samples for the model. The model is trained with the cross-entropy loss function.

$$L = -\sum_{G \in g \cup g'} (y_G log(\sigma(c)) + (1 - y_G)\frac{log((1 - \sigma(c)))}{k}) \quad (5)$$

Among them, G represents a triplet containing positive and negative samples, and y_G values are 0 and 1, which represent the label of the triple. σ is $softmax$ function, c represents the fully connected layer output of the final part of the model, so $\sigma(c)$ represents the classification score of the triplet. k is a positive negative sample proportion and its value is 3 in our experiment.

TABLE II
THE STATISTICS OF DATASETS.

Dataset	$\|\mathcal{E}\|$	$\|\mathcal{R}\|$	Train	Valid	Test
Wiki27K	27122	62	74,793	20,242/1994	20,244/1994
FB15K-237-N	13,104	93	87,282	14,082/2046	16,452/2048

III. Experiments

A. Datasets and Evaluation Metrics

We employ two sub-datasets from Wikidata and Freebase. These datasets constitute the benchmark dataset of the PKGC model [6], which is built on the basis of Wikidata and manually annotated with real negative triples to form a new dataset named Wiki27K.

The FB15K-237 dataset contains many mediator (CVT) nodes, which result in Cartesian product relations that do not make sense for the corresponding prediction tasks and inappropriately increase the accuracy of the model [12]. To address this issue, [6] created a new dataset called FB15K-237-N by removing the relations containing mediator nodes in FB15K-237. Table II shows details of the datasets. The numbers in the right two columns represent the number of triples used for evaluation under the closed world assumption (CWA) and open world assumption (OWA), respectively.

We evaluate the performance of KGC models on two tasks: link prediction and triple classification. The most commonly used indicators for link prediction are MRR and Hits@n (n is 1, 3, 10), and the final results are the average values obtained by replacing the head entity and tail entity, respectively.

The triple classification task aims to determine whether a given triple is correct, making it essentially a binary classification task. Therefore, accuracy and F1 score are used to evaluate the performance of triple classification. Our model is implemented using PyTorch based on the PKGC, with bert-base-cased used as the PLM.

B. *Baselines*

To establish a baseline for our study, we select some representative models, including TransE [3], TransC [13], ConvE [14], RotatE [4], KG-BERT [5], MTL-KGC [8], StAR [14], and PKGC [6]. When compared to PLM-based models, the original StAR and PKGC models, which use the RoBERTa-Large [16], have a large number of parameters and a high negative sampling ratio, leading to long training times. To ensure a more fair comparison of model performance, we modified the StAR and PKGC models to use bert-base-cased as PLM for the experiment, labeled as StAR (BERT-base) and PKGC (BERT-base), respectively.

C. *Main Results and Analysis*

Table III presents the link prediction results of our model and the baseline models on the datasets of Wiki27K and FB15K-237-N. Table IV lists the comparative results of these models on the triple classification task under the two datasets. All metrics are multiplied by 100. Based on the experimental results, the following conclusions can be drawn:

(1) Comparing the results of the link prediction and triple classification experiments, we observe that, in the PLM-based models, the direct concatenation of entities and relations is less effective than the method of transforming triples into coherent sentences through templates. This suggests that the input format of the data has a significant impact on the model's performance. As PLMs are trained on natural language text, input data that is coherent sentences are consistent with the training data to some extent. This enables better utilization of the implicit knowledge of PLMs, resulting in sentence embedding representations that contain rich semantic information.

(2) In the comparative experiments against these baseline models, our model achieved the best results. This indicates that both textual semantic information and structural information are crucial for KGC tasks. In comparison to the StAR, our model's use of pre-trained structural embeddings can effectively solve the problem of mismatch between the fine-tuning speed of the BERT and the training speed of the KGE model, leading to desirable results.

(3) Compared to the PLM-based models, our model achieves desirable results by using BERT-base as PLM. This suggests that the method of transforming triples into coherent sentences can better utilize the implicit knowledge in PLMs, and performance improvement can be achieved by incorporating entity definitions and learnable tokens. Other models may require larger PLMs to obtain textual semantic information. Additionally, compared to the method of using the $[CLS]$ vector directly to represent the semantic information of triples for classification, the feature extraction method by extracting features from the entire sentence embedding matrix using a feature extractor can achieve more effective results. This may be because the $[CLS]$ vector does not pay enough attention to the semantics of entities and relations, while the local features obtained through CNNs focus more on the semantics of entities and relations, thereby better completing the task of KGC.

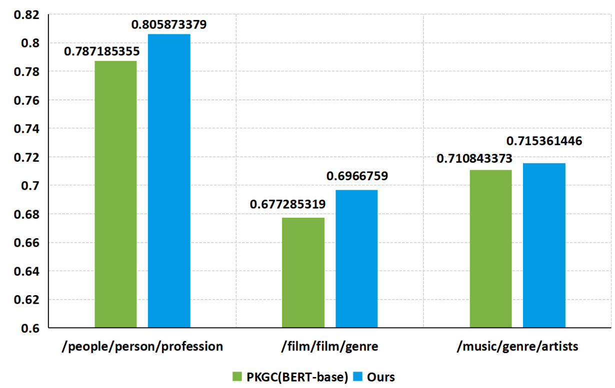

Fig. 2. The comparison of the prediction accuracy rate of the top three relations corresponding to the number of samples.

(4) Compared to KGE models, PLM-based models have more advantages in triple classification tasks. This suggests that incorporating PLMs into KGC models can better determine the correctness of triples, possibly because such models use cross-entropy loss functions during training, making them more proficient in classification tasks. In addition, the results under the OWA and the CWA in triple classification tasks do not differ significantly, which may be due to the smaller ratio of negative triples containing errors obtained through negative sampling compared to link prediction tasks.

D. *Case Study*

Table V shows the results of some triple classification prediction cases in the FB15K-237-N dataset. We selected cases where our model predicted correctly while PKGC (BERT-base) predicted incorrectly and analyzed them. The label represents the correctness of the triple, where 1 indicates a positive triple that exists in the knowledge graph, and 0 indicates a negative triple obtained through negative sampling. It is apparent that in the same prediction task, our model outperforms PKGC (BERT-base) in relations that require comprehensive structural information, such as the "/film/film/country" relation.

To further explore the performance of our model on various types of relations, we conducted a detailed comparative analysis experiment on the triple classification task in the FB15K-237-N dataset. Figure 2 presents the prediction results of the model on the top three relations with the corresponding sample numbers of 2622, 1444, and 1328, respectively. The values in the figure represent the accuracy of the predictions. It can be observed that our model has more advantages in resolving complex relation patterns (1-N, N-1, and N-N), such as "/people/person/profession" and "/film/film/genre".

E. *Ablation experiment*

Our model incorporates learnable tokens in the data processing part. To explore the impact of this component on the model, we conducted a corresponding ablation experiment. The input data of the comparison model had the $[SPECIAL]$ tokens removed. The comparison results are shown in Figure 3.

TABLE III

LINK PREDICTION RESULTS ON WIKI27K AND FB15K-237-N.

Model	Wiki27K				FB15K-237-N			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
TransE[a]	15.5	3.2	22.8	37.8	25.2	15.2	30.1	45.9
TransC[a]	17.5	12.4	21.5	33.9	23.3	12.9	29.8	39.5
ConvE [a]	22.6	16.4	24.4	35.4	27.3	19.2	30.5	42.9
RotatE[a]	21.6	12.3	25.6	39.4	27.9	17.7	32.0	**48.1**
KG-BERT[a]	19.2	11.9	21.9	35.2	20.3	13.9	20.1	40.3
MTL-KGC [a]	21.7	13.8	23.5	37.9	24.8	15.5	25.6	43.6
StAR (BERT-base)	21.1	13.4	24.0	35.0	24.7	15.8	23.9	40.1
PKGC (BERT-base)	26.2	20.3	28.4	39.5	29.2	21.7	31.8	45.5
SS-KGC (Ours)	**26.9**	**21.0**	**29.2**	**40.0**	**30.1**	**22.7**	**32.7**	46.3

[a]The results are taken from [6]. The best result is in bold.

TABLE IV

TRIPLET CLASSIFICATION RESULTS ON WIKI27K AND FB15K-237-N.

Model	Wiki27K		FB15K-237-N	
	ACC	F1	ACC	F1
TransE[a]	65.5/64.2	72.3/71.5	66.2/71.5	71.7/70.4
TransC[a]	68.7/68.4	71.5/71.2	66.4/64.6	71.3/70.8
ConvE[a]	70.7/68.8	73.5/73.5	67.3/67.3	71.8/73.7
RotatE[a]	72.3/64.0	75.1/71.3	67.9/63.2	72.3/69.9
KG-BERT[a]	83.7/82.4	84.3/83.1	71.8/72.7	72.8/73.6
MTL-KGC[a]	84.3/83.6	85.1/84.4	73.8/74.4	73.0/74.5
PKGC (BERT-base)	87.4/87.1	87.2/86.9	76.5/76.0	76.7/75.8
SS-KGC (Ours)	**88.0/88.4**	**88.2/88.5**	**77.1/78.1**	**77.2/78.4**

[a]The results are taken from [6]. The best result is in bold.

TABLE V

CASES OF TRIPLET CLASSIFICATION UNDER THE FB15K-237-N DATASET.

head entity	relation	tail entity	label
/m/03gfvsz	/broadcast/content/artist	/m/01k_mc	1
/m/0fb7sd	/film/film/country	/m/09c7w0	1
/m/0cq806	/film/film/language	/m/04h9h	1
/m/024lt6	/film/film/film_festivals	/m/0kfhjq0	1
/m/0kfhjq0	/film/film/genre	/m/06nbt	1
/m/04rqd	/broadcast/content/artist	/m/012z8_	0
/m/0cwy47	/film/film/country	/m/0cwy47	0
/m/01f6x7	/film/film/language	/m/064_8sq	0
/m/0gg5qcw	/film/film/film_festivals	/m/0bmj62v	0
/m/0bth54	/film/film/genre	/m/07s9rl0	0

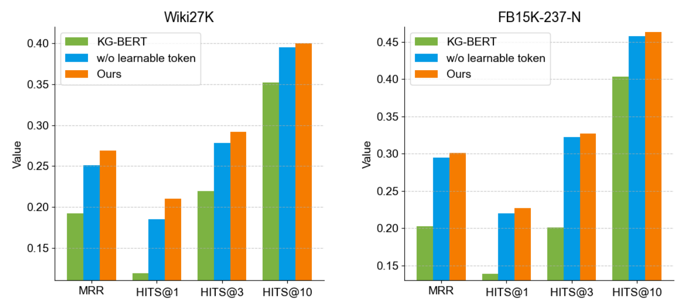

Fig. 3. Comparison of ablation experiments on learnable tokens.

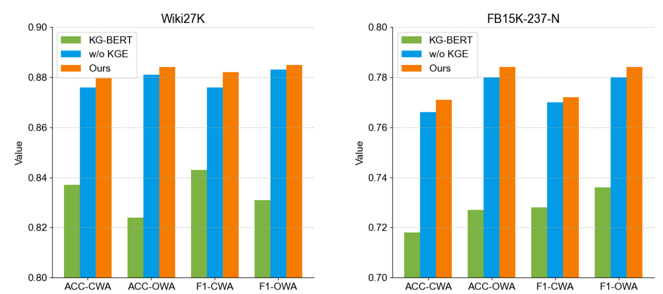

Fig. 4. Comparison of ablation experiments on structure embedding.

It shows that after removing the learnable tokens, the model's performance drops. However, compared to other models (such as KG-BERT) that use the direct concatenation of entities and relations, our model still demonstrates very competitive results. This indicates that the form of the input data is crucial in exploiting the implicit knowledge of the PLM. Additionally, experiments demonstrate that the learnable tokens can effectively enhance the model's performance, enabling the full mining of semantics from the triple text and better utilization of the implicit knowledge of the PLM.

To explore the impact of structural embeddings on model performance, we conducted ablation experiments on the triple classification task and analyzed the results by comparing them. Figure 4 shows that the model's performance declines on the triple classification task without structural embeddings, whether based on the OWA or the CWA, demonstrating the significant impact of structural information on KGC tasks. It's worth noting that our model without KGE still demonstrates very competitive results compared to other baseline models.

We also found that using CNN in our model yields better results across all metrics compared to using MLP.

IV. RELATED WORK

A. Pre-trained Language Models

Pre-trained language models (PLMs) are a type of natural language processing technique based on deep learning, typically trained on large amounts of unlabeled text data [7], [16]. These models can learn the grammar, semantics, and contextual information in natural language, generating meaningful language representations. PLMs can be applied to various natural language processing (NLP) tasks, such as text classification, named entity recognition, and machine translation, through fine-tuning or transfer learning. In recent years, PLMs have become one of the research hotspots in the field of NLP, achieving many significant experimental results.

B. Knowledge graph completion Models

Knowledge graph completion is one of the important research in the field of knowledge graph [1]. A series of methods have been proposed for KGC. Among them, methods based on representation learning have received widespread attention. It includes KGE models and PLM-based models [15]. These models aim to learn low-dimensional representations for entities and relations in the knowledge graph, enabling efficient reasoning and prediction of missing links.

The most classic KGE models are the translation-based KGE models [3]. They represent each entity and relation as a low-dimensional vector and then regard the relation as the distance between the head entity and the tail entity [3], [4], [17], [18]. These models are popular because of their simplicity and efficiency.

Compared to translation-based models, semantic matching models reflect the confidence of the semantic information of the triple. It includes bilinear models and neural network models. Bilinear models use a bilinear matrix to measure the similarity between entities and relations [19], [20], while neural network models use various neural network architectures to capture the complex semantics of entities and relations [14], [21].

PLM-based KGC models are a new emerging method developed in recent years. The main idea of this model is to use the PLM to convert the textual descriptions of entities and relations into vector representations, and then use these vectors to complete missing entities or relations, such as KG-BERT [5], MTL-KGC [8], StAR [15], and PKGC [6].

V. Conclusion

In this paper, we propose a KGC model that combines text semantic and structural embedding to improve the performance of KGC. The model uses BERT to encode text and incorporates learnable tokens and relations templates to construct coherent sentences from triple labels and entity definitions. Our model also uses multi-head attention to transform the semantic space of the embeddings of entities and relations obtained by the KGE model. Finally, the model uses CNN to extract features from the matrices spliced by two types of information. Experimental results show that our model outperforms other baseline models on link prediction and triplet classification tasks, demonstrating its effectiveness in capturing both structural and semantic information in KGs.

Acknowledgements

This research was supported by the School of Film Xiamen University - Dongbo Future Artificial Intelligence Research Institute (Xiamen) Co., Ltd. Joint Laboratory for created the Metaverse (School Agreement No. 20223160C0026), School of Film Xiamen University - Xiaozhi Deep Art (Xiamen) Artificial Intelligence Research Institute Co., Ltd. Computational Art Joint Laboratory (School Agreement No. 20213160C0032), and School of Information Xiamen University - Xiamen Yinjiang Smart City Joint Research Center (School Agreement No. 20213160C0029).

References

[1] S. Ji, S. Pan, E. Cambria, P. Marttinen, and S. Y. Philip, "A survey on knowledge graphs: Representation, acquisition, and applications," *IEEE transactions on neural networks and learning systems*, vol. 33, no. 2, pp. 494–514, 2021.

[2] H. Zhou, T. Young, M. Huang, H. Zhao, J. Xu, and X. Zhu, "Commonsense knowledge aware conversation generation with graph attention." in *IJCAI*, 2018, pp. 4623–4629.

[3] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," *Advances in neural information processing systems*, vol. 26, 2013.

[4] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "Rotate: Knowledge graph embedding by relational rotation in complex space," *arXiv preprint arXiv:1902.10197*, 2019.

[5] L. Yao, C. Mao, and Y. Luo, "Kg-bert: Bert for knowledge graph completion," *arXiv preprint arXiv:1909.03193*, 2019.

[6] X. Lv, Y. Lin, Y. Cao, L. Hou, J. Li, Z. Liu, P. Li, and J. Zhou, "Do pre-trained models benefit knowledge graph completion? a reliable evaluation and a reasonable approach." Association for Computational Linguistics, 2022.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[8] B. Kim, T. Hong, Y. Ko, and J. Seo, "Multi-task learning for knowledge graph completion with pre-trained language models," in *Proceedings of COLING*, 2020, pp. 1737–1743.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[10] Y. Chen, "Convolutional neural network for sentence classification," Master's thesis, University of Waterloo, 2015.

[11] X. Han, W. Zhao, N. Ding, Z. Liu, and M. Sun, "Ptr: Prompt tuning with rules for text classification," *AI Open*, vol. 3, pp. 182–192, 2022.

[12] F. Akrami, M. S. Saeef, Q. Zhang, W. Hu, and C. Li, "Realistic re-evaluation of knowledge graph completion methods: An experimental study," in *Proceedings of ACM SIGMOD*, 2020, pp. 1995–2010.

[13] X. Lv, L. Hou, J. Li, and Z. Liu, "Differentiating concepts and instances for knowledge graph embedding," *arXiv preprint arXiv:1811.04588*, 2018.

[14] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2d knowledge graph embeddings," in *Proceedings of AAAI*, vol. 32, no. 1, 2018.

[15] B. Wang, T. Shen, G. Long, T. Zhou, Y. Wang, and Y. Chang, "Structure-augmented text representation learning for efficient knowledge graph completion," in *Proceedings of WWW*, 2021, pp. 1737–1748.

[16] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[17] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proceedings of AAAI*, vol. 28, no. 1, 2014.

[18] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proceedings of AAAI*, vol. 29, no. 1, 2015.

[19] B. Yang, W.-t. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," *arXiv preprint arXiv:1412.6575*, 2014.

[20] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *Proceedings of ICML*. 2016, pp. 2071–2080.

[21] S. Vashishth, S. Sanyal, V. Nitin, N. Agrawal, and P. Talukdar, "Interacte: Improving convolution-based knowledge graph embeddings by increasing feature interactions," in *Proceedings of AAAI*, vol. 34, no. 03, 2020, pp. 3009–3016.