Bilateral Masking with prompt for Knowledge Graph Completion

Yonghui Kong[1], **Cunhang Fan**[1*], **Yujie chen**[1], **Shuai Zhang**[2],
Zhao Lv[1], **Jianhua Tao**[2*]
[1]Anhui Province Key Laboratory of Multimodal Cognitive Computation,
School of Computer Science and Technology, Anhui University, Hefei 230601, China,
[2]Department of Automation, Tsinghua University, China,
E22201093@stu.ahu.edu.cn cunhang.fan@ahu.edu.cn E22201148@stu.ahu.edu.cn
zhang_shuai@mail.tsinghua.edu.cn kjlz@ahu.edu.cn jhtao@tsinghua.edu.cn

Abstract

The pre-trained language model (PLM) has achieved significant success in the field of knowledge graph completion (KGC) by effectively modeling entity and relation descriptions. In recent studies, the research in this field has been categorized into methods based on word matching and sentence matching, with the former significantly lags behind. However, there is a critical issue in word matching methods, which is that these methods fail to obtain satisfactory single embedding representations for entities.To address this issue and enhance entity representation, we propose the Bilateral Masking with prompt for Knowledge Graph Completion (BMKGC) approach.Our methodology employs prompts to narrow the distance between the predicted entity and the known entity. Additionally, the BMKGC model incorporates a bi-encoder architecture, enabling simultaneous predictions at both the head and tail. Furthermore, we propose a straightforward technique to augment positive samples, mitigating the problem of degree bias present in knowledge graphs and thereby improving the model's robustness. Experimental results conclusively demonstrate that BMKGC achieves state-of-the-art performance on the WN18RR dataset.

1 Introduction

Knowledge graphs (KGs) are graph-structured knowledge bases, typically consisting of triples, denoted as (h, r, t), where h represents the head entity, r represents the relation, and t represents the tail entity. Prominent examples of KGs include Freebase , Wikidata(Vrandečić and Krötzsch, 2014), YAGO(Suchanek et al., 2007),ConceptNet(Speer et al., 2017), and WordNet(Miller, 1992). KGs find applications in various domains, including information retrieval(Xiong et al., 2017), recommendation systems(Huang et al., 2018), and question answering(Sun et al., 2019a). However, KGs encounter the challenge of incompleteness as real-world information continuously evolves Figure1. Hence, the tasks of knowledge graph completion (KGC)(Galárraga et al., 2017) hold significant importance. In recent years, to enhance KG completion and utilization, significant research efforts have been devoted to the field of knowledge embedding (KE), which aims to map KGs into low-dimensional vector spaces. Existing

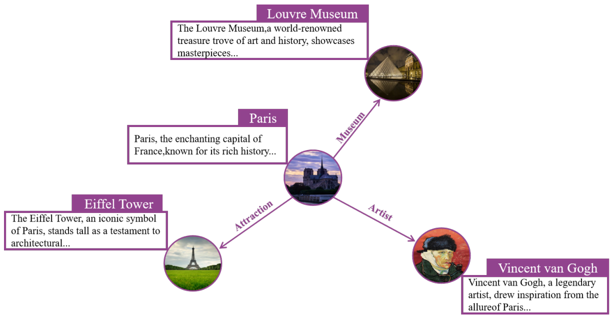

Figure 1: Example of multiple facts in a KG.Every entity is associated with a unique name and corresponding textual descriptions.

knowledge embedding methods fall into two categories: structure-based methods and description-based methods. Structure-based methods, such as TransE(Bordes et al., 2013), RotatE(Sun et al., 2019b) and TuckER(Balazevic et al., 2019), leverage the topological information within the knowledge graph to represent entities and relations. Description-based methods can be further categorized into sentence matching and word matching approaches.The methods utilized in this paper draw inspiration from sentence matching techniques while falling under the category of word matching methods. These methods enrich entity representations by incorporating descriptive information related to entities and relations. They transform entity-relation pairs and their descriptions into natural language-style sequences and uti-

*Corresponding authors

Findings of the Association for Computational Linguistics: NAACL 2024, pages 240–249
June 16-21, 2024 ©2024 Association for Computational Linguistics

lize pre-trained language models (PLM) such as BERT(Devlin et al., 2019)) to encode them. This encoding process generates comprehensive representations of entities and relations.

The distinction between the two approaches lies in their treatment of triplets.Sentence matching transforms KGC into a text matching task, as seen in SimKGC(Wang et al., 2022a), enabling description-based methods to surpass structure-based methods for the first time.In contrast, word matching involves predicting the masked positions in the input data and matching them with all candidate entities. Prior word matching techniques, like C-LMKE(Wang et al., 2022b), have commonly employed a specific token, frequently denoted as 'ent_id', as a replacement for the original entity while handling candidate entities. As a consequence, the suboptimal representation of candidate entities significantly affects the overall performance.Furthermore, they position the description of the known entity immediately following the entity, inevitably widening the gap between the predicted entity and the known entity. Consequently, this introduces additional challenges for the model in predicting unknown entities.These two factors have led to a significant disparity between word matching methods and sentence matching methods.

In this paper, inspired by SimKGC(Wang et al., 2022a), we introduce a new fuzzy operation to handle relations of candidate entities. We achieve this operation by simultaneously making predictions in both the head and tail encoders. Moreover,we design prompts to narrow the distance between the known entity and the predicted entity, with the aim of strengthening their association. The objective of this fuzzy operation is to improve the word matching method based on descriptive embedding, resulting in enhanced single entity embedding representations. Furthermore, graph-related tasks commonly encounter the issue of degree bias(Kojaku et al., 2021), where nodes with lower degrees tend to exhibit weaker representations and poorer downstream performance.We take inspiration from SimCSE(Gao et al., 2021) and utilize the dropout mechanism of pre-trained language models to acquire additional positive samples, addressing the degree bias issue in knowledge graphs. We conduct experiments on three popular benchmark datasets, including WN18RR and FB15k-237. According to automated evaluation metrics MRR and Hits@k ($k \in \{1, 3, 10\}$), our Bilateral Masking with prompt for Knowledge Graph Completion (BMKGC) method achieves state-of-the-art performance on the WN18RR dataset and competitive results on the FB15k-237 dataset.

2 Related Work

The knowledge graph is composed of triples, denoted as (h, r, t), where each triple consists of a head entity, $h \in \mathcal{E}$, connected to a tail entity, $t \in \mathcal{E}$, through a relationship, $r \in \mathcal{R}$. Presently, knowledge graph completion approaches involve representing entities and relationships as vectors in a lower-dimensional space, known as knowledge graph embedding. These knowledge graph embedding are further categorized into structure-based and description-based embedding.

Structure-Based Knowledge Embedding We categorize structural-based knowledge embedding into three types. Firstly, the first type relies on translation-based techniques, such as TransE(Bordes et al., 2013), TransH(Wang et al., 2014), and TransR(Lin et al., 2015), which employ distance-based scoring functions. These methods generate embeddings for the head entity (h), relationship (r), and tail entity (t) based on specific translations, allowing assessment of the plausibility of triples (h, r, t) using distance scoring functions.Secondly, the ComplEx(Trouillon et al., 2016) model utilizes factorization and complex embeddings to enhance the representation of entities and relationships. TuckER(Balazevic et al., 2019) considers knowledge graph completion as a 3-D binary tensor decomposition problem and explores the effectiveness of various factorization techniques.Thirdly, knowledge graph embedding is approached as a deep learning task, utilizing various neural network architectures. ConvE(Dettmers et al., 2018) adopts convolutional neural networks (CNNs) to capture interactions between entities and relations, while CompGCN(Vashishth et al., 2020) improves knowledge graph representations by incorporating multi-layered information through graph convolutional networks (GCNs)(Schlichtkrull et al., 2018). Simultaneously, HittER(Chen et al., 2021) and CoKE(Wang et al., 2019) leverage Transformers to process information within the knowledge graph. These models demonstrate innovative methodologies based on different neural network architectures aimed at advancing the performance and precision of knowledge graph completion.

Description-Based Knowledge Embedding Recent studies have utilized pre-trained language models (PLM) such as BERT(Devlin et al., 2019) to improve the completion of knowledge graphs (KGs) by converting incomplete triples into natural language inputs. In this paper, we categorize descriptive-based embedding into two types: sentence matching and word matching. Sentence matching involves dividing the triple (h, r, t) into the head entity h, connected by a relationship r, and the tail entity t. The goal is to determine the most plausible triple by evaluating the similarity of the semantic meanings between these two components. SimKGC(Wang et al., 2022a) significantly enhances performance by introducing efficient contrastive learning. On the other hand, word matching methods such as MEM-KGC(Choi et al., 2021) mask the tail entity and consider the head entity and relationship as context for predicting the masked entity. C-LMKE(Wang et al., 2022b) utilizes predictions for unknown tail entities from the input, together with representations of all entities within the same batch, to obtain the most probable tail entity.

Degree Bias The C-LMKE(Wang et al., 2022b) enhances the representations of long-tail entities in KGs by incorporating degrees and leveraging text information, effectively improving the performance of PLM on KGs. While KG-Mixup(Shomer et al., 2023) validates the presence of degree bias in embedding-based Knowledge Graph Completion (KGC) and identifies its main factors, it is worth noting that KG-Mixup only focuses on enhancing structural-based embedding methods. Consequently, there is a lack of data augmentation techniques for descriptive-based embedding methods to enhance KG completion performance for long-tail entities. Our proposed BMKGC model aims to address this gap.

3 Methods

This section provides a comprehensive introduction to BMKGC, as depicted in Figure2. Firstly, we present a concise definition of knowledge graphs and their relevance to link prediction. Subsequently, we elucidate the principles and implementation specifics of our Bidirectional masking technique. Finally, a succinct overview of degree compensation is provided.

3.1 Definitions and Notation

A knowledge graph is a directed graph consisting of entities and relationships, denoted as $\mathcal{G} = \{\mathcal{E},\ \mathcal{R},\ \mathcal{T}\}$, where $\mathcal{E}$ represents the set of entities, $\mathcal{R}$ denotes the set of relationships, and $\mathcal{T}$ stands for the set of triples, defined as $T = \{(h, r, t) \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}\}$. BMKGC maximizes the utilization of descriptive information related to entities and relationships, denoted as d_h, d_r, d_t for h, r, and t, respectively. This additional information serves as input to PLM, enabling them to understand entities and relationships and acquire their embedding representations. Link prediction tasks aim to predict missing parts of triples within existing knowledge graphs. Using the widely adopted entity ranking evaluation protocol, predicting the tail entity (h,r,?) involves ranking all entities given h and r, and similarly, for predicting the head entity (?,r,t). In this study, we reverse the relationship by transforming (h, r, t) into (t, r^{-1}, h). This reversal allows us to solely focus on predicting the tail entity.

3.2 Bidirectional Masking

Knowledge Graph Completion (KGC) tasks initially aimed to predict missing entities given specific entities and relationships, similar to Masked Language Modeling (MLM)(Devlin et al., 2019) where certain words within input text sequences are randomly masked, and the model predicts these masked words. Motivated by MLM, our proposed BMKGC model adopts a similar approach, employing a bi-encoder architecture . We initialize two encoders, named $BERT_{head}$ and $BERT_{tail}$, with the same pre-trained language model but operate independently without parameter sharing.

Given a specific triplet $T = (h, r, t)$ along with corresponding descriptions d_h, d_r, and d_t, the first encoder, $BERT_{head}$, is utilized to predict the missing tail entity based on the provided head entity h, its description d_h, and relationship r along with its description d_r. To closely align with the triplet structure, we utilize a prompt format input as follows:

$$x_{base} = [CLS](h, r, [MASK]), d_h[SEP] \quad (1)$$

Our objective with this prompt is to strengthen the connection between the predicted entity and the known entity relationship. Unlike previous methods that placed [MASK] at the end and interleaved the description of the head entity within it, resulting in the predicted entity being too distant from

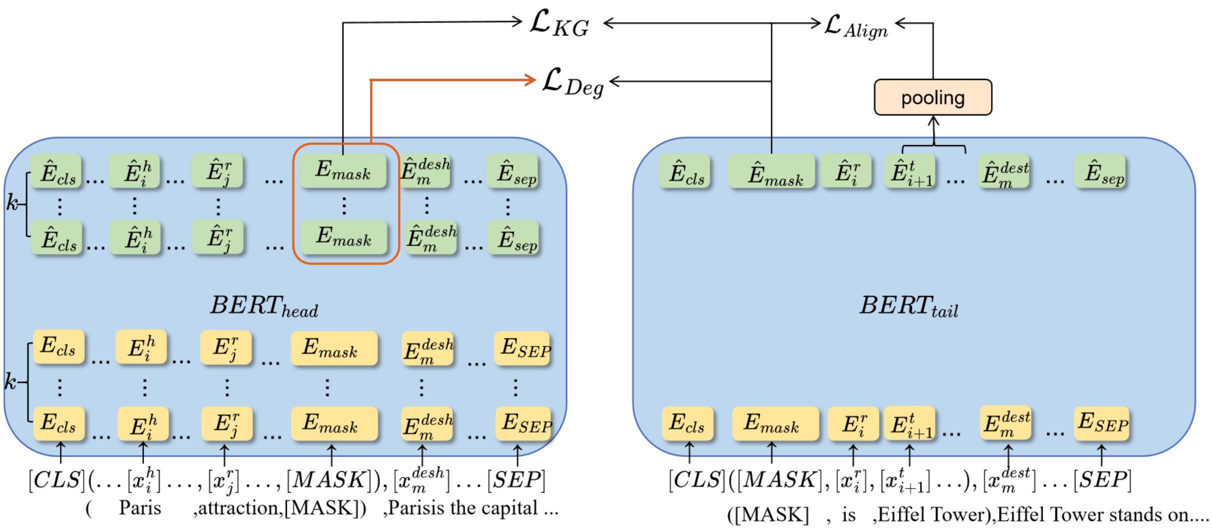

Figure 2: The architecture of BMKGC.We will illustrate the usage of a triplet (Paris, attraction, Eiffel Tower) as an example. The prompt input is depicted at the bottom of the figure, where the input in $BERT_{head}$ is duplicated k times to mitigate the inherent bias in the knowledge graph (KG). $BERT_{tail}$ input incorporates [MASK], enabling the implementation of the proposed bilateral masking technique.

the head entity within the sentence, we place our prompt in a more coherent position. x_{base} is then fed into $BERT_{head}$, yielding the representation of the [MASK] token e_p as follows:

$$e_p = BERT_{head}(x_{base})_{mask} \tag{2}$$

The second encoder, $BERT_{tail}$, performs a similar operation on candidate tail entities and their descriptions for entity representation. Similar to $BERT_{head}$, we adopt a prompt format and introduce a new relation "is". The input format is as follows:

$$x_{tail} = [CLS]([MASK], is, t), d_t[SEP] \tag{3}$$

This relation "is" signifies an equivalence relationship, allowing $BERT_{tail}$ to predict the tail entity based on entity t and its description d_t, aiming to obtain a robust single embedding representation for the tail entity through self-prediction.We refer to it as a fuzzy operation. x_{tail} is then fed into $BERT_{tail}$, generating the representation of the MASK token as follows:

$$e_t = BERT_{tail}(x_{tail})_{mask} \tag{4}$$

Finally, we calculate the similarity between e_p and e_t using cosine similarity. Additionally, to employ contrastive learning, we utilize an InfoNCE loss function(Chen et al., 2020), where the $\mathcal{L}_{KG}$ loss is defined as follows:

$$score = cos(e_p, e_t) = \frac{e_p \cdot e_t}{||e_p||\,||e_t||} \tag{5}$$

$$\mathcal{L}_{KG} = InfoNCE(score, L) \tag{6}$$

where L represents the true label of the training dataset. In order to prevent the embeddings obtained from the fuzzy operation from excessively deviating from the original word meanings, we apply an $\mathcal{L}_{align}$ loss to minimize the distance between e_t and the original word representation, given by:

$$\mathcal{L}_{Align} = InfoNCE(cos(e_t, e_w), L) \tag{7}$$

Here, e_w denotes the embedded representation obtained by averaging the original tail entity, and L is the true label of the training dataset.

3.3 Degree compensation

Based on the research conducted in KG-Mixup(Shomer et al., 2023), it has been observed that the performance of the KGC task is influenced by the in-degree of tail entities. Taking inspiration from Mixup(Zhang et al., 2018), they generated supplementary positive samples, which were previously absent from the training dataset, to augment entities with lower degrees. This approach effectively mitigated the degree bias issue in structure-based methods. However, there is currently a lack of methods to generate similar additional positive samples in description-based approaches.Drawing inspiration from SimCSE(Gao et al., 2021), a method utilizing the dropout mechanism in BERT to generate similar yet distinct sentences as positive samples, we propose a similar approach. By incorporating a contrastive learning framework, we effectively capture textual representations, resulting in a significant enhancement of performance in text matching tasks. In our method, we calculate the degrees of all entities within the training dataset, establish a threshold value η and repeatedly pass the corresponding head entities and relationships of tail entities with degrees below this threshold to $BERT_{head}$ k times.

$$\tilde{\mathcal{T}} = \begin{cases} \mathcal{T}_{train} \cup \{(\tilde{h}, \tilde{r}, t)\}_{i=1}^{k} & d_{tail} < \eta, \\ \mathcal{T}_{train} & else, \end{cases} \tag{8}$$

where d_{tail} represents the in-degree of the entity with t as the tail, $(h, r, t) \in \mathcal{T}_{train}$ represents the original training triples, $(\tilde{h}, \tilde{r}, t)$ is a synthetic sample, and $\tilde{\mathcal{T}}_{v,r}$ denotes the new set of triples used during training.

It is important to note that tail entities are included only once, consistent with the KG-Mixup research. Here, k represents the difference between the tail entity's degree and the threshold value η. These repeated input-output samples are considered additional positive samples, providing some mitigation for the issue of degree bias. The loss term $\mathcal{L}_{Deg}$ aligns with the previous KG loss (6). The final loss of BMKGC is the weighted sum of the losses from each task. We experimentally determine the weight λ for $\mathcal{L}_{Align}$ as demonstrated below:

$$\mathcal{L} = \mathcal{L}_{KG} + \lambda \mathcal{L}_{Align} + \mathcal{L}_{Deg} \tag{9}$$

4 Experiments

4.1 Experimental Setup

Dataset	N_e	N_r	N_{Train}	N_{Valid}	N_{Test}
WN18RR	40943	11	86835	3034	3134
FB15K-237	14541	237	272115	17535	20466

Table 1: Statistics of the datasets.

Datasets Our experiments were conducted on two datasets, namely WN18RR and FB15k-237. Both FB15k-237 and WN18RR datasets are knowledge graph datasets utilized for relation prediction tasks. FB15k-237 was extracted from Facebook's Freebase data and comprises a comprehensive collection of entities and relations. In contrast, WN18RR is based on the WordNet knowledge graph, encompassing entities such as nouns and verbs, and their corresponding relations. The primary distinctions between these two datasets lie in their data sources and the quantity of relations. FB15k-237 is derived from the real-world Freebase knowledge graph, encompassing a wide array of entities and relations. Conversely, WN18RR is specifically oriented towards the domain of natural language processing, encompassing more precise and restricted entities and relations. Additionally, FB15k-237 has undergone a reduction in redundant and inconsistent relations to enhance the overall data quality. The statistical data is shown in Table1. In our experiments, the state-of-the-art SimKGC(Wang et al., 2022a) model is adopted as the baseline. We compare our method with a set of structure-based methods, namely TransE(Bordes et al., 2013) , RotatE(Sun et al., 2019b), ,ConvE(Dettmers et al., 2018), and CompGCN(Vashishth et al., 2020). Additionally, we compare our method with description-based approaches, including KG-BERT(Yao et al., 2019), MTL-KGC(Kim et al., 2020), C-LMKE(Wang et al., 2022b),KGLM(Youn and Tagkopoulos, 2022),LP-BERT(Li et al., 2022) and StAR(Wang et al., 2021).

Evaluation Metrics The evaluation metrics for the link prediction task involve adopting an entity ranking approach based on previous research. For each tested triple (h, r, t), the prediction of the tail entity t entails ranking all entities using the provided head entity h and relationship r. A similar process is employed for predicting the head entity. Four automatic evaluation metrics are utilized, namely Mean Reciprocal Rank (MRR) and Hits@k (k $\in \{1, 3, 10\}$). MRR represents the average reciprocal rank of all tested triples, while Hits@k calculates the proportion of correctly ranked entities within the top k positions. The reported MRR and Hits@k values are obtained using the filtered setting(Bordes et al., 2013), which involves the common practice of excluding other correct entities (that also form triples in the knowledge graph) from the list. The measurements are calculated by averaging in both directions: head entity prediction and tail entity prediction. In general, a good model is expected to achieve higher MRR and Hits@k values, as well as a lower MR.

Hyperparameters To enhance the performance of pre-trained language models, we employ bert-base-uncased (English) to initialize our encoder. During training, we conduct training sessions on WN18RR and FB15k-237 datasets using a batch size of 512 on A800 GPU for 50 and 5 epochs,

Method	WN18RR				FB15k-237			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
structure-based methods								
TransE(Bordes et al., 2013)	24.3	4.3	44.1	53.2	27.9	19.8	37.6	44.1
RotatE(Sun et al., 2019b)	47.6	42.8	49.2	57.1	33.8	24.1	37.5	53.3
ConvE(Dettmers et al., 2018)	43.0	40.0	44.0	52.0	32.5	23.7	35.6	50.1
CompGCN(Vashishth et al., 2020)	47.9	44.3	49.4	54.6	**35.5**	**26.4**	**39.0**	**53.5**
description-based methods								
KG-BERT(Yao et al., 2019)	21.6	4.1	30.2	52.4	-	-	-	42.0
MTL-KGC(Kim et al., 2020)	33.1	20.3	38.3	59.7	26.7	17.2	29.8	45.8
C-LMKE(Wang et al., 2022b)	61.9	52.3	67.1	78.9	30.6	21.8	33.1	48.4
KGLM(Youn and Tagkopoulos, 2022)	46.7	33.0	53.8	74.1	28.9	20.0	31.4	46.8
LP-BERT(Li et al., 2022)	48.2	34.3	56.3	75.2	31.0	22.3	33.6	49.0
StAR(Wang et al., 2021)	40.1	24.3	49.1	70.9	29.6	20.5	32.2	48.2
SimKGC(Wang et al., 2022a)	66.5	58.6	71.6	80.0	33.5	24.9	36.2	51.0
BMKGC(ours)	**66.9**	**59.0**	**72.0**	**80.7**	33.2	24.7	36.5	51.4

Table 2: Main results for WN18RR and FB15K-237 datasets. The best result for each metric and each KGE method is shown in bold.

respectively.In our study, we solely considered the samples within each batch as negative samples. The learning rates are set to 9×10^{-5} and 1×10^{-5}. Moreover, we initialize the temperature τ to 0.05 and set the margin value to 0.02. In optimizing our model, we utilize the AdamW optimizer with linear learning rate decay.The weight λ in the loss function is assigned a value of 0.5.

4.2 Main Results

We utilized the experimental results from StAR(Wang et al., 2021) for embedding-based methods, and we obtained the results of SimKGC(Wang et al., 2022a) through its corresponding code. The optimal outcomes for other models were extracted from the experimental data presented in their respective papers. In Table2, our proposed BMKGC demonstrated significant advancements in all metrics for the WN18RR dataset, reaching a state-of-the-art level. The Hits@10 metric has demonstrated a notable enhancement to 80.7, surpassing the robustness of all prior models. This improvement can be credited to the utilization of a fuzzy operation that effectively portrays entities and acquires superior individual embedding representations. Consequently, it empowers the model to achieve a more comprehensive comprehension of entities with lengthier names, thereby augmenting the overall robustness of the model.Moreover, the Hits@1 metric has also increased to 59.0, indicating enhanced precision. This improvement can be ascribed to our prompt design, which leverages the closer proximity between known entities and predicted entities, resulting in more accurate predictions made by the model.

When comparing BMKGC to the baseline SimKGC on the FB15k-237 dataset, it performs better in Hits@3 and Hits@10 but exhibits a decrease in other metrics. However, our performance in FB15k-237 is better than other description-based methods.We believe this is primarily attributed to two factors. Firstly, unlike other datasets, FB15k-237 contains a lower number of entities (14,541) and relations (237), resulting in a denser graph structure with an average degree of approximately 37 for each entity. This suggests the presence of multiple relationships per entity, which are more intricate and cannot be simply explained by a single word but require the combination of multiple words. This complexity poses a challenge for the encoder as it struggles to comprehend and process such multilayered relationships. Furthermore, (Cao et al., 2021) remarked that numerous connections in the FB15k-237 dataset cannot be predicted based on the available information.

5 Analysis

Prompt In contrast to previous methods that rely on descriptions, our prompt aims to reduce the distance between the predicted tail entity, the head entity, and their relationship. We hypothesize that this approach strengthens the association between the entities, thereby improving the model's ability to make description-based predictions. To vali-

w/ prompt	(Paris,Attraction,[MASK])Paris:the enchanting capital of France,known for its rich history... ([MASK],is,Eiffel Tower)Eiffel Tower: an iconic symbol of Paris,stands tall as a testament to architectural...)
w/o prompt	(Paris:the enchanting capital of France,known for its rich history...,Attraction,[MASK]) ([MASK],is,Eiffel Tower: an iconic symbol of Paris, stands tall as a testament to architectural...)
w/ pooling	Paris:the enchanting capital of France,known for its rich history...,Attraction,[MASK] Eiffel Tower: an iconic symbol of Paris, stands tall as a testament to architectural...
w/ token_id	ent_id_i:the enchanting capital of France,known for its rich history...,Attraction,[MASK] ent_id_j: an iconic symbol of Paris, stands tall as a testament to architectural..

Table 3: Various methods employ distinct input formats.

date this hypothesis, we designed a control groups with specific inputs $(h, d_h, r, [MASK])$. Table 3 illustrates the input formats for the prompts we proposed in the $BERT_{head}$ and $BERT_{tail}$ models in the first row, while the second row shows the input format without prompts.Keeping all other parameters consistent, we conducted experiments on the WN18RR dataset. In Table4,the results clearly demonstrate that the performance on each metric is significantly lower when compared to our proposed prompt. This provides substantial evidence supporting the effectiveness of BMKGC.Designing prompts in this manner offers an additional advantage by effectively addressing the max_length constraint of the tokenizer. In the absence of prompts, there would be a necessity to predefine the length of entity descriptions. However, this approach presents a risk of potentially excluding the [MASK] token when the descriptions exceed the maximum input length defined by the tokenizer. Moreover, establishing a predetermined description length in advance is deemed as detrimental to the model's performance.

	MRR	Hits@1	Hits@3	Hits@10
w/ prompt	66.9	59.0	72.0	80.7
w/o prompt	62.3	54.9	66.3	75.9

Table 4: Analysis of prompt on the WN18RR dataset.

Bilateral Masking We have implemented a novel methodology for representing candidate entities, which distinguishes itself from previous descriptive-based methods by introducing a fuzzy operation to manipulate the predicted forms of the candidate tail entities. Table 3 presents the application of average pooling for entities in the third row, while the last row showcases the substitution of the original entities with special token_id. In this instance, i and j correspond to the entity id of Paris and Eiffel Tower, respectively, in the dataset. Table 5 reveals that our proposed BMKGC method outperforms the other two approaches across all metrics, exhibiting particularly substantial enhancements in Hits@1. These findings indisputably establish the effectiveness of our BMKGC model. Additionally, we observe that Average Pooling underperforms, suggesting its inability to replicate the pooling technique employed in sentence-based matching methods within word-based matching approaches. This limitation negatively impacts the representation of words, directly diminishing the model's performance.

Method	MRR	Hits@1	Hits@3	Hits@10
MASK	66.9	59.0	72.0	80.7
Pooling	62.3	53.0	68.0	79.1
token_id	64.7	55.9	70.2	80.3

Table 5: Analysis of approach for handling entities on the WN18RR dataset. MASK is the bilateral masking method proposed by us. Pooling is the average pooling applied to candidate entities. Token_id is used to replace the original entity name with the entity reference ID

Degree bias In Figure3, we analyze the performance of BMKGC and BMKGC without additional positive samples on WN18RR. The x-axis represents the degree of the tail entities that require completion, which is derived from the training set. We categorize them into four groups since entities with a degree exceeding one hundred are not within the scope of the degree bias problem in our research. The y-axis represents the mean reciprocal rank (MRR) of each group. By conducting a comparison, it is evident that the inclusion of extra positive samples enhances the MRR for entities with lower degrees, corroborating the effectiveness

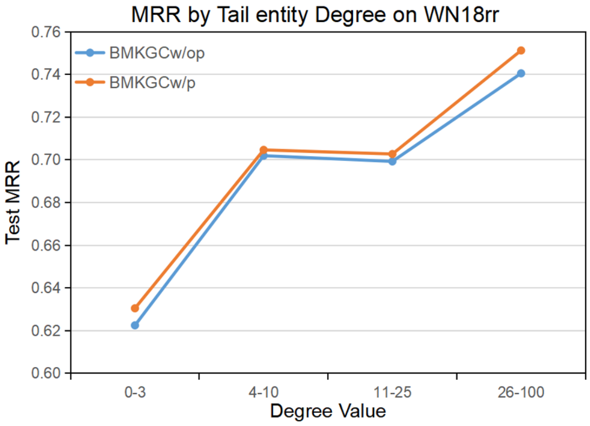

Figure 3: Analysis of using degree compensation different degree groups in terms of MRR on the WN18RR dataset.

of our proposed method in mitigating the degree bias issue to some extent.

Alignment loss ration We conducted an investigation into the potential constraining effect of alignment loss on the prediction of $BERT_{tail}$. In Table6, we employed an annotation scale λ with values ranging from 0 to 1, including 0, 0.25, 0.5, 0.75, and 1. It was observed that both excessively small and large values of λ adversely impacted the model's prediction accuracy. In the case of a small λ, the predictions made by $BERT_{tail}$ deviated from the true meaning of the word and exhibited an undue bias towards the expected answer. Conversely, a large λ value weakened the representation of entities. As a result of our analysis, we selected λ=0.5 as it yielded the best overall performance.

Loss Ration	MRR	Hits@1	Hits@10
$\lambda = 0$	65.8	57.8	80.3
$\lambda = 0.25$	66.4	58.3	80.5
$\lambda = 0.5$	66.9	59.0	80.7
$\lambda = 0.75$	66.4	58.4	80.1
$\lambda = 1$	66.1	57.8	80.3

Table 6: Performance comparison on the WN18RR dataset across the different loss ratio .

6 Conclusion

This paper proposes BMKGC, a method that effectively enhances entity representation by predicting candidate entities during training to obtain improved single embedding representations. Furthermore, we propose a simple method to increase positive samples, thus alleviating the issue of degree bias in the knowledge graph. Extensive experimental results convincingly demonstrate that our approach achieves state-of-the-art performance. In future research, we will concentrate on exploring entity-related information within the PLM, reducing the impact of noise generated during pretraining, and further advancing entity representation.

Limitations

Our proposed method optimizes the representation of entities by using bilateral masking and prompts to enhance the model's prediction. Furthermore, we introduce dropout as a means to mitigate the degree bias issue in the knowledge graph. Our method demonstrates significant performance improvements; however, it comes with associated costs in terms of time and computational resources. In the $BERT_{head}$ section, the positions of [MASK] in the input are stored and later retrieved during output processing due to varying entity lengths. Moreover, the inclusion of additional positive samples slightly increases the computational resources requirement of our model compared to the one without them. Nonetheless, these costs are deemed acceptable in practice. Moreover, our research did not explore hrad negative samples beyond those within the batch. We consider this as a potential future research direction, aiming to delve into more challenging negative samples and enhance entity representation.

Acknowledgements

This work is supported by the STI 2030—Major Projects (No.2021ZD0201500), the National Natural Science Foundation of China (NSFC) (No.62201002), Distinguished Youth Foundation of Anhui Scientific Committee (No.2208085J05), Special Fund for Key Program of Science and Technology of Anhui Province(No.202203a07020008), Open Fund of Key Laboratory of Flight Techniques and Flight Safety, CACC(No.FZ2022KF15) and the Open Research Projects of Zhejiang Lab (NO.2021KH0AB06).

References

Ivana Balazevic, Carl Allen, and Timothy Hospedales. 2019. TuckER: Tensor factorization for knowledge graph completion. 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 5185–5194, Hong Kong, China. Association for Computational Linguistics.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2787–2795.

Yixin Cao, Xiang Ji, Xin Lv, Juanzi Li, Yonggang Wen, and Hanwang Zhang. 2021. Are missing links predictable? an inferential benchmark for knowledge graph completion. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6855–6865, Online. Association for Computational Linguistics.

Sanxing Chen, Xiaodong Liu, Jianfeng Gao, Jian Jiao, Ruofei Zhang, and Yangfeng Ji. 2021. HittER: Hierarchical transformers for knowledge graph embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10395–10407, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Bonggeun Choi, Daesik Jang, and Youngjoong Ko. 2021. Mem-kgc: Masked entity model for knowledge graph completion with pre-trained language model. *IEEE Access*, 9:132025–132032.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 1811–1818. AAAI Press.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Luis Galárraga, Simon Razniewski, Antoine Amarilli, and Fabian M. Suchanek. 2017. Predicting completeness in knowledge bases. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM 2017, Cambridge, United Kingdom, February 6-10, 2017*, pages 375–383. ACM.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y. Chang. 2018. Improving sequential recommendation with knowledge-enhanced memory networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pages 505–514. ACM.

Bosung Kim, Taesuk Hong, Youngjoong Ko, and Jungyun Seo. 2020. Multi-task learning for knowledge graph completion with pre-trained language models. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1737–1743, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Sadamori Kojaku, Jisung Yoon, Isabel Constantino, and Yong-Yeol Ahn. 2021. Residual2vec: Debiasing graph embedding with random graphs. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 24150–24163.

Da Li, Ming Yi, and Yukai He. 2022. Lp-bert: Multi-task pre-training knowledge graph bert for link prediction. *ArXiv preprint*, abs/2201.04843.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, pages 2181–2187. AAAI Press.

George A. Miller. 1992. WordNet: A lexical database for English. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*.

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.

Harry Shomer, Wei Jin, Wentao Wang, and Jiliang Tang. 2023. Toward degree bias in embedding-based knowledge graph completion. In *Proceedings of the ACM Web Conference 2023*, pages 705–715.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 4444–4451. AAAI Press.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*, pages 697–706. ACM.

Haitian Sun, Tania Bedrax-Weiss, and William Cohen. 2019a. PullNet: Open domain question answering with iterative retrieval on knowledge bases and text. 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 2380–2390, Hong Kong, China. Association for Computational Linguistics.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019b. Rotate: Knowledge graph embedding by relational rotation in complex space. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2071–2080. JMLR.org.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha P. Talukdar. 2020. Composition-based multi-relational graph convolutional networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Bo Wang, Tao Shen, Guodong Long, Tianyi Zhou, Ying Wang, and Yi Chang. 2021. Structure-augmented text representation learning for efficient knowledge graph completion. In *Proceedings of the Web Conference 2021*, pages 1737–1748.

Liang Wang, Wei Zhao, Zhuoyu Wei, and Jingming Liu. 2022a. SimKGC: Simple contrastive knowledge graph completion with pre-trained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4281–4294, Dublin, Ireland. Association for Computational Linguistics.

Quan Wang, Pingping Huang, Haifeng Wang, Songtai Dai, Wenbin Jiang, Jing Liu, Yajuan Lyu, Yong Zhu, and Hua Wu. 2019. Coke: Contextualized knowledge graph embedding. *ArXiv preprint*, abs/1911.02168.

Xintao Wang, Qianyu He, Jiaqing Liang, and Yanghua Xiao. 2022b. Language models as knowledge embeddings. *ArXiv preprint*, abs/2206.12617.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada*, pages 1112–1119. AAAI Press.

Chenyan Xiong, Russell Power, and Jamie Callan. 2017. Explicit semantic ranking for academic search via knowledge graph embedding. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, pages 1271–1279. ACM.

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

Jason Youn and Ilias Tagkopoulos. 2022. Kglm: Integrating knowledge graph structure in language models for link prediction. *ArXiv preprint*, abs/2211.02744.

Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.