Multilingual Knowledge Graph Completion from Pretrained Language Models with Knowledge Constraints

Ran Song[1,2], Shizhu He[3,4], Shengxiang Gao[1,2],
Li Cai[5], Kang Liu[3,4], Zhengtao Yu[1,2] [*] and **Jun Zhao[3,4]**

[1] Faculty of Information Engineering and Automation,
Kunming University of Science and Technology, Kunming, China
[2] Yunnan Key Laboratory of Artificial Intelligence, Kunming, China
[3] The Laboratory of Cognition and Decision Intelligence for Complex Systems,
Institute of Automation, Chinese Academy of Sciences, Beijing, China
[4] School of Artificial Intelligence, University of Chinese Academy of Science, Beijing, China
[5] Meituan, Beijing, China
{song_ransr}@163.com, {shizhu.he,kliu,jzhao}@nlpr.ia.ac.cn,
caili03@meituan.com, {gaoshengxiang.yn,ztyu}@hotmail.com

Abstract

Multilingual Knowledge Graph Completion (mKGC) aim at solving queries like $(h, r, ?)$ in different languages by reasoning a tail entity t thus improving multilingual knowledge graphs. Previous studies leverage multilingual pretrained language models (PLMs) and the generative paradigm to achieve mKGC. Although multilingual pretrained language models contain extensive knowledge of different languages, its pretraining tasks cannot be directly aligned with the mKGC tasks. Moreover, the majority of KGs and PLMs currently available exhibit a pronounced English-centric bias. This makes it difficult for mKGC to achieve good results, particularly in the context of low-resource languages. To overcome previous problems, this paper introduces global and local knowledge constraints for mKGC. The former is used to constrain the reasoning of answer entities, while the latter is used to enhance the representation of query contexts. The proposed method makes the pretrained model better adapt to the mKGC task. Experimental results on public datasets demonstrate that our method outperforms the previous SOTA on Hits@1 and Hits@10 by an average of 12.32% and 16.03%, which indicates that our proposed method has significant enhancement on mKGC.

1 Introduction

Knowledge graphs are collections of entities and facts, and utilized as a valuable resource in a variety of natural language processing (NLP) tasks, such as Question Answering and Recommender Systems (Shah et al., 2019; Du et al., 2021; Wang et al., 2019). The language-specific nature of many NLP tasks necessitates to consider the knowledge

Figure 1: The top part introduces unbalance language distribution for DBpedia. The low part shows the sampling comparison results of Prix-LM model and our method. The type of prediction entity and the correct answer are shown in brackets and **red** font, respectively. Our approach exhibits superior consistency and accuracy in generating answers.

expressed in a particular language. For example, multilingual question answering needs multilingual knowledge graphs (Zhou et al., 2021). The utilization of multilingual knowledge graphs (mKGs) with a vast amount of knowledge in multiple languages, such as DBpedia (Lehmann et al., 2015), Wikidata (Vrandečić and Krötzsch, 2014), can be advantageous in plenty of NLP tasks (Zhou et al., 2021; Fang et al., 2022).

There is a significant amount of potential facts that have not been captured in current knowledge graphs, resulting in their incompleteness (Chen et al., 2020). To address this issue, various stud-

---

* Corresponding author

ies have proposed for Knowledge Graph Completion (KGC) to automatically discovery potential facts through observed facts (Bordes et al., 2013), rules (Meilicke et al., 2019) and language models (Lv et al., 2022).

In fact, as shown in Figure 1, there is more English-centric knowledge than other languages, so that it is difficult to leverage knowledge graphs on non-English tasks. For example, English-centric commonsense reasoning tasks obtain better development and performance than other languages (Lin et al., 2021a). And the knowledge coverage of non-English knowledge graphs is even worse, it will poses challenges for traditional KGC methods to achieve superior performance.

Nowadays, pretrained language models (PLMs) learn various knowledge modeling capabilities (Petroni et al., 2019; Jiang et al., 2020) from massive unlabeled data. And most studies have demonstrated that the knowledge contained within PLMs can significantly improve the performance of downstream tasks (Li et al., 2021; Lin et al., 2021b). Most recently, Prix-LM (Zhou et al., 2022) approached mKGC as an end-to-end generative task using multilingual PLMs. For example, for predicting the missing entity of the query *(86th All Japan Football Championship, Stadium, ?)* (see Figure 1), Prix-LM converts the query into a sequence with pre-defined template, which is then processed by an encoder to generate a query representation. The decoder then uses this representation to generate the final answer *Axis Bird Stadium*.

Despite the successes achieved through the combination of PLMs and the generative paradigm, there remain limitations for mKGC. On the one hand, the gap between the pretraining task and the KGC task may contribute to the limitations. It arise that the answers generated by Prix-LM are ambiguous in type. On the other hand, languages and tokens that occur more frequently in the pretraining data have richer representations. Linguistic bias for KGs and PLMs would arise that entities in low-resource languages are difficult to be represented, resulting answer incorrect. As illustrated in Figure 1, the query *(86th All Japan Football Championship, stadium, ?)* expects a response of the type *stadium*, but the top-ranked answers from Prix-LM are diverse, and the top answer is incorrect.

We argue that the incorporation of knowledge constraints into the generation process can increase PLMs suitability for mKGC tasks. We categorize

knowledge effective for mKGC into global and local knowledge. Global Knowledge limit the types of answers based on building the relationship of entity and relation representations. This helps to ensure that the generated answers are semantically and logically consistent with the intent of query. On the other hand, local knowledge in PLMs can enhance the ability to comprehend the interconnections between the sub-tokens within the query. This helps the model to better understand the context of query and generate more accurate answers. Incorporating knowledge constraints into the generative process brings two advantages for mKGC: 1) It makes PLMs to better adapt to mKGC task. 2) It enables PLMs to learn more effective representations from low-resource data.

In this paper, we propose to incorporate the global and local knowledge into the answer generation process through two knowledgeable tasks. To learn global knowledge, special tokens ([H],[R],[T]) are introduced as semantic representations of head entity, relation, and tail entity in a triple. A scoring function measures the plausibility of the resulting facts, such as $||h_{[H]} + h_{[R]} - h_{[T]}||_{L_{1/2}}$. Since the same special token is used in each triple in different languages, trained models are able to learn knowledge reasoning ability beyond language boundaries. To capture local knowledge, we consider the representation of answer and each word of query as two separate distributions $P(H_q)$ and $P(H_{[T]})$, and then use an estimator to estimate and maximize the mutual information between them $I(H_q; H_{[T]})$. The local knowledge serves to augment the query representations for trained model through the utilization of minimal amounts of data. The experimental results on seven language knowledge graph from DBpedia show that our proposed method achieves significant improvement as compared to Prix-LM and translated-based methods. We publicize the dataset and code of our work at https://github.com/Maxpa1n/gcplm-kgc.

In short, our main contributions are as follows:

- We attempt to utilize diverse knowledge constraints to enhance the performance of PLM-based mKGC. It effectively addresses the inconsistency of PLM and mKGC task, and alleviates language and data bias from PLMs and KGs.

- Our proposed method can enrich query representation and enhance answer generation by
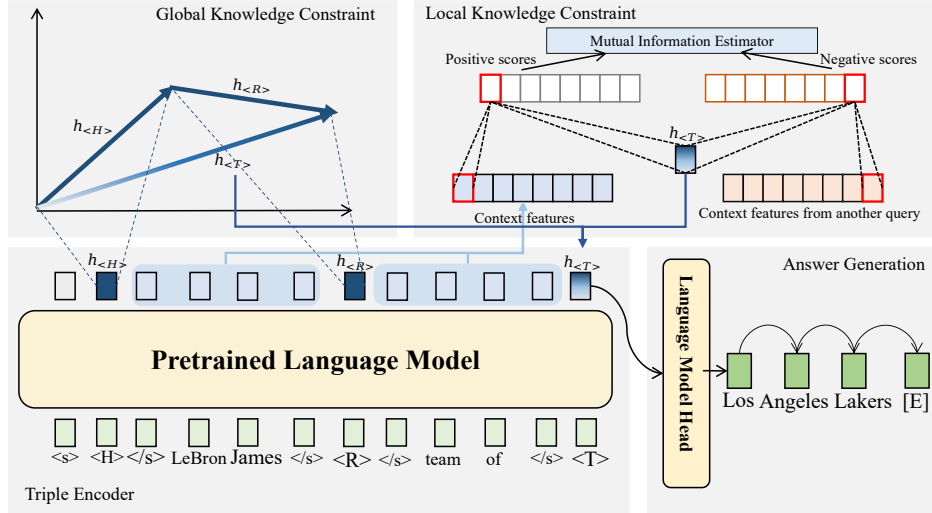
Figure 2: This figure illustrates the architecture of the complete model, which is composed of four main components: a query encoder, a global knowledge constraint, a local knowledge constraint, and an answer generation module. The global knowledge learn from representations of head and relation (navy blue). The local knowledge learn from representations of query words (light blue). We use different colors to represent entities and relation in each module for a triple.

introducing global knowledge constraints for entity placeholders and mutual information constraints for other contextual symbols.

- Our proposed method outperforms the Prix-LM (Zhou et al., 2022) in both mKGC and cross-lingual entity alignment, as shown by experiments on a public dataset. The performance of our method on Hits@1, Hits@3, and Hits@10 shows an average improvement of 12.32%, 11.39%, and 16.03%, respectively.

2 Basic Model

A knowledge graph $\mathcal{G} = (\mathcal{R}, \mathcal{E})$ is a collection of connected information about entities, often represented using triples (h, r, t) where $r \in \mathcal{R}$ is relation and $h, t \in \mathcal{E}$ are entities. Prix-LM is an important work of mKGC and is also used as the basic model in this paper. Prix-LM transfer link prediction from discriminative task to generative task for mKGC. The goal of mKGC is to generate the missing tail entity, which may contain multiple tokens, for the query $(h, r, ?)$ of different languages. The use of template is employed as a means of transforming queries into textual sequences that can be encoded by PLMs. The template includes special tokens, which serve to identify the specific role of each element within the query triple:

<s>[H]X_h</s></s>[R]X_r</s></s>[T]X_t[E]

where <s> is beginning token of sentence and </s> is the separator, both are applied in PLMs, as known as [CLS] and [SPE]. [H], [R] and [E] are additional special tokens for the representation of head, relation and tail. [E] is the end-of-sequence token. $X_h \in \{\mathbf{x}_1^h, \mathbf{x}_2^h, \mathbf{x}_3^h, ..., \mathbf{x}_n^h\}$ are text words of head entity, X_r and X_t in the same way.

The training goal is to generate the tail entity X_t by giving the sequence containing the head entity X_h and relation X_r. For example, for the query *(LeBron James, team member of, ?)*, the constructed sequence is <s>[H] *LeBron James*</s></s>[R] *team member of*</s></s>[T], and the target of mKGC is generate *Los Angeles Lakers* [E]. The process is as follows:

$$P_\theta(X_t|X_h, X_r) = \prod_{x_i \in X_t}^{x_i} p(x_i|x_{<i}, \theta) \quad (1)$$

where θ is the pretrained model parameter. According to the mechanism of causal language model, the probability of i-th token depend on previous token representation $\mathbf{h}_{i-1}$:

$$p(x_i|X_{<i}) = \text{softmax}(\mathbf{W}\mathbf{h}_{i-1}) \quad (2)$$

where W is causal language model decoder from PLMs.

The utilization of PLMs for generating answers directly can be subject to language bias, resulting in ambiguous and incorrect answers. The representation of the special token $[T]$ is a crucial factor in

determining the quality of the generated answers. To improve the representation of the $[T]$ token, we have implemented two supplementary strategies aimed at incorporating additional knowledge into its representation.

3 The Proposed Model

In this section, we describe the components of our proposed approach. The architecture of the model is depicted in Figure 2. Our approach comprises four key components: a query encoder, a global knowledge constraint, a local knowledge constraint, and an answer generation module. These components operate in tandem to generate accurate and coherent answers for given queries.

3.1 Triple Encoder

We leverage the PLM to encode the triple and an attention mask to control the access to each subtoken in the sequence during training process. We use previous template to convert a triple (h, r, t) to a sequence $S_{(h,r,t)} \in \{X_h, X_r, X_t, X_a\}$, and X_a is special token. The attention mask mechanism allows the query sequence to be seen as the source text and the answer entity as the target text. The process as following:

$$\text{PLM}(S_{(h,r,t)}) = H \tag{3}$$

where hidden representation of triple is $H \in \{\mathbf{h}^{[H]}, \mathbf{h}_1^h, .., \mathbf{h}^{[R]}, \mathbf{h}_1^r, ..., \mathbf{h}^{[T]}, \mathbf{h}_1^t, ..., \mathbf{h}^{[E]}\}$. The attention mask is a matrix that specifies whether each subtoken should be attended or ignored, as illustrated in Figure 3. By making special tokens only visible to their own subtokens, model can effectively separate each role in a triple. And the mask matrix $\mathbf{M}$ add in attention score calculated by query $\mathbf{Q}$, key $\mathbf{K}$, value $\mathbf{V}$:

$$\mathbf{M} = \begin{cases} 0, & \text{allow to attend} \\ -\infty, & \text{prevent from attending} \end{cases} \tag{4}$$

$$\mathbf{A} = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}} + \mathbf{M})\mathbf{V} \tag{5}$$

where $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{l \times d}$, l is length of the input sequence, and d is the hidden size.

3.2 Global Knowledge Constraint

To bridge the gap between the pretraining task and the KGC task, we introduce the global knowledge build logical relationship between entities. Unlike

previous approaches such as Prix-LM, our method does not rely on cross-lingual links for equivalent entities to learn shared knowledge in different languages. Instead, shared knowledge between languages is learned through the global knowledge constraint, which is inspired by embedding-based methods. We leverage the TransE framework in our model, and methods such as CompleX, RotatE are also applicable. The goal of the global knowledge constraint is to represent entities and relation in a semantic space and enforce the translational principle: $h + r \approx t$:

$$\|\mathbf{h}_{[H]} + \mathbf{h}_{[R]}\| = \|\mathbf{h}_{[T]}\| \tag{6}$$

where $\mathbf{h}_{[H]}$, $\mathbf{h}_{[R]}$, $\mathbf{h}_{[T]}$ are special tokens representation, and $\|.\|$ is L1 norm. And a triple global knowledge score is described by:

$$score(h, r, t) = \|\mathbf{h}_{[H]} + \mathbf{h}_{[R]} - \mathbf{h}_{[T]}\| \tag{7}$$

We use the same special tokens for different languages. The following loss function is used to optimize the model.

$$\mathcal{L}_p = \sum_{g_j \in G}^{g_j} \sum_{(h,r,t)_i \in g_j}^{(h,r,t)_i} (score(h_i, r_i, t_i) + \gamma) \tag{8}$$

where G is all language knowledge graphs set, and γ is correction factor.

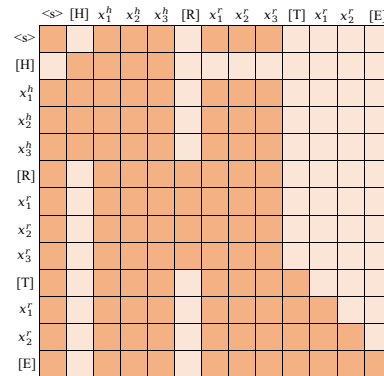Figure 3: The operation mechanism of mask matrix during training process. The darker squares indicate that attention is allowed, while the lighter squares indicate that attention is suppressed.

3.3 Local Knowledge Constraint

The local knowledge enables the model to learn more accurately for generated answers with low-resource data. Therefore, we consider establishing the connection between query and answer in

a triple. Specifically, we view the the representation of query words H_q and tail entity $H_{[T]}$ as two distributions and maximizing the mutual information between them $I(H_q, H_{[T]})$. The theoretical foundation for this idea is provided by MIEN (Belghazi et al., 2018), which demonstrates that mutual information follows a parametric lower-bound:

$$I(H_q; H_{[T]}) \geq \hat{I}_\theta(H_q; H_{[T]}) \qquad (9)$$

Inspired from previous Mutual Information Maximization (Tschannen et al., 2019; Zhang et al., 2020) (MIM) method in unsupervised learning, we take the local features, represented by H_q, and the global features, represented by $H_{[T]}$, as the inputs for MIM. Benefit from the mask mechanism and PLM's powerful learning capability, we do not strictly distinguish the parameter of encoder and decoder different from previous works. In this work, we select a Jensen-Shannon MI estimator to parameterize Mutual Information:

$$\begin{aligned}
\hat{I}_\theta^{(JSD)}(H_q, H_{[T]}) := \\
E_{\mathbb{P}}[-sp(T_\theta(H_q, H_{[T]}))] \qquad (10) \\
-E_{\mathbb{P}X\widetilde{\mathbb{P}}}[sp(T_\theta(H_q', H_{[T]}))]
\end{aligned}$$

where $H_q \in \{\mathbf{h}_1^h, \ldots, \mathbf{h}_m^h, \mathbf{h}_1^r, \ldots, \mathbf{h}_n^r\}$ is query words representation, m is head entity length, n is relation length. $H_{[T]} \in \{\mathbf{h}_{[T]}\}$ is tail entity representation. T_θ is a discriminator function support by the PLM parameters. H_q' is representation sampled from other query in the same min batch. And $\mathbb{P} = \widetilde{\mathbb{P}}$ make guarantee the expectation easy to calculated. $sp(x) = log(1 + e^x)$ is the softplus activation function. The learning object is to make PLM estimate and maximize the Mutual Information:

$$\theta = \underset{\theta}{\operatorname{argmax}} \frac{1}{|G|} \sum_{b_j \in G}^{b_j} \hat{I}_\theta^{(JSD)}(H_q^j, H_{[T]}^j) \quad (11)$$

where b_j is mini batch from training dataset. To optimize model by gradient descent, we set loss function as following:

$$\mathcal{L}_E = \sum_{b_j \in G}^{b_j} (E_{\widetilde{\mathbb{P}}}^j - E_{\mathbb{P}}^j) \qquad (12)$$

where the $E_{\mathbb{P}}^j$ is expectation for query and tail entity. The local knowledge constraint within PLM enhance its capacity to obtain rich representations of queries and tail entities, particularly in situations where training data is limited.

3.4 Answer Generation Module

Follow the paradigm that given a serialized query and generate answer token, we use the casual language model with PLM. The generation loss function as Cross Entropy Loss function:

$$\mathcal{L}_G = \sum_{(h,r,t)_i \in G}^{i} \sum_{x_j \in X_i^t}^{x_j} x_j log(f(x_{<j})) \qquad (13)$$

where the $f(\cdot)$ is like Formula 2, x_j is subtoken of tail entity.

In training process, the model would generate answer with global and local knowledge, we define the loss for model as:

$$\mathcal{L} = \mathcal{L}_G + \alpha \mathcal{L}_P + \beta \mathcal{L}_E \qquad (14)$$

where α and β are hyperparameter. The mask mechanism achieved that all subtokens of tail entity be trained in one round.

3.5 Inference

During the inference phase of our model, we utilize an autoregressive approach to generate the tokens of tail entity for given query. This autoregressive approach involves predicting the next token based on the previous tokens. The query $(h, r, ?)$ be transferred to a sequence X_q and generating the answer entity by trained model. The process as following:

$$x_i = \underset{x_i}{\operatorname{argmax}} P(x_i | X_q \cap x_1 \cdots \cap x_{i-1}) \qquad (15)$$

where $x_i \in X_t$. Additionally, we assume a closed-world setting and utilize constrained beam search to restrict the final output to a predefined set of possibilities, in order to ensure the validity of the generated answer.

4 Experiments

In this section, we evaluate the effectiveness of our approach on tasks related to mKGC and Entity Alignment for mKGs. To further understand the role of the various knowledge-gathering strategies in our method, we also conduct ablation experiments. Additionally, we provide case studies to demonstrate the superior performance of our method on specific examples. These experiments and analyses provide insight into the strengths and limitations of our approach for addressing challenges in mKGC for sparse knowledge graphs.

	MODEL	DE	FI	FR	HU	IT	JA	TR	AVG
	TransE	0.00	0.01	0.02	0.03	0.04	0.02 s	0.06	0.02
	ComplEx	4.09	2.45	2.50	3.28	2.87	2.41	1.00	2.65
Hits@1	RotatE	6.72	5.87	8.40	16.27	6.91	6.21	6.85	8.17
	Prix-LM (Single)	12.86	19.81	18.01	28.72	16.21	19.81	23.79	19.88
	Prix-LM	14.32	18.78	16.47	29.68	14.32	18.19	21.57	19.04
	Ours	**17.54**	**20.74**	**18.34**	**30.91**	**14.98**	**22.05**	**25.20**	**21.39**
	TransE	6.14	6.54	6.60	14.91	5.95	7.22	8.20	7.93
	ComplEx	8.47	5.28	5.19	6.70	4.31	4.68	2.11	5.24
Hits@3	RotatE	10.52	7.42	14.62	21.75	12.11	9.75	11.29	12.49
	Prix-LM (Single)	23.09	28.75	24.75	38.44	25.32	29.02	33.05	28.91
	Prix-LM	23.68	29.54	23.15	39.80	25.46	27.01	31.45	28.58
	Ours	**30.40**	**29.74**	**26.36**	**44.18**	**27.03**	**30.79**	**35.48**	**31.99**
	TransE	17.54	17.80	15.26	29.00	14.16	20.65	19.35	19.10
	ComplEx	9.35	8.21	8.91	16.96	8.76	8.23	5.24	9.38
Hits@10	RotatE	14.61	8.61	19.49	28.31	18.48	14.44	17.13	17.29
	Prix-LM (Single)	33.82	38.91	34.04	47.31	36.61	38.81	38.50	38.28
	Prix-LM	33.91	41.29	32.25	46.23	35.18	36.12	37.50	37.49
	Ours	**41.81**	**43.44**	**35.15**	**58.00**	**39.15**	**42.45**	**44.55**	**43.50**

Table 1: In this table, the results of seven language-specific knowledge graph completion (KGC) tasks are presented. The embedding-based methods, including TransE, complEx, and RotatE, were implemented using the OpenKE framework (Han et al., 2018). The results for these methods were obtained by training separate knowledge graphs for each language. The Single make the monolingual version, which is trained independently for each language. The numbers in **bold** represent the best results among the methods and languages considered.

		DE	FI	FR	HU	IT	JA	TR	AVG
	Entity	39,842	36,892	106,955	27,765	86,988	68,279	29,120	56,549
Training	Relation	1,544	945	2,358	999	1,539	2,542	1,008	1,562
	Triple	27,014	28,040	83,940	24,193	66,904	50,164	24,013	43,467
	Entity	501	766	2,452	988	2,240	1,206	749	1,272
Validation	Relation	122	142	362	142	257	303	101	204
	Triple	264	435	1,407	614	1,286	671	432	730
	Entity	649	916	2,687	1,154	2,499	1,414	822	1,449
Testing	Relation	135	147	377	154	271	322	95	214
	Triple	342	511	1,559	731	1,461	789	496	841
T/E Ratio		0.69	0.79	0.81	0.92	0.80	0.76	0.76	0.79

Table 2: The table show the statistics of multi language knowledge graph dataset. The T/E Ratio is equal to the number of triples divided by the number of entities.

4.1 Datasets and Evaluation Metrics

To evaluate our method, we utilize the Link Prediction dataset provided by Prix-LM (Zhou et al., 2022) and split it by the closed-world setting. The dataset consists of data from DBpedia, a large multilingual knowledge graph, and the amount of data is shown in Table 2. We ensure that entities and relations appearing in the validation and test sets are included in the training set. We introduce a ratio between entities and triples as a measure of the knowledge density of the dataset. This ratio has a lower bound of 0.5, which indicates that there are no cross-links between triples. The ratio of our dataset is much lower than that of publicly available datasets. The evaluation metrics we use are standard Hits@1, Hits@3, and Hits@10, which are commonly used in the evaluation of KGC methods.

4.2 Implementation Details

In our experiments, we used XLM-R (Base) as the base pre-trained language model and did not introduce any additional parameters beyond those provided by XLM-R. The model was implemented using the Huggingface Transformers library (Wolf et al., 2020) and the hyperparameters α and β were set to 0.001 and 0.005. The learning rate and batch size were selected from the sets {4e-5, 5e-5} and {128, 256}. And the maximum length of a triple sequence was 35. The model was trained using a single Nvidia RTX 3039 GPU.

4.3 Multilingual Knowledge Graph Completion

Our method for mKGC was compared to various embedding-based methods and Prix-LM on seven languages KG, as shown in Table 1. The results show that our method outperformed Prix-LM on the metrics of Hits@1, Hits@3, and Hits@10, with average improvements of 12.32%, 11.39%, and 16.03%, respectively. These improvements suggest that the integration of both global and local knowledge significantly enhances the effectiveness of the mKGC task, leading to a higher ability to accurately predict missing triple in KG.

It is worth noting that the low knowledge density in the training set can hinder the performance of traditional embedding-based methods, which rely on the presence of sufficient training data to learn

	model	zh-km	zh-th	zh-lo	zh-my
H1	Prix-LM	65.02	22.42	62.15	41.56
	Ours	**67.25**	**24.07**	62.15	**43.49**
H3	Prix-LM	68.37	26.87	64.27	46.44
	Ours	**69.23**	**27.65**	**64.80**	**47.65**
H10	Prix-LM	70.15	**30.50**	66.12	49.08
	Ours	70.15	30.35	**66.27**	**50.73**

Table 3: This table presents the results of entity alignment tasks for low-resource languages. The parallel entity pairs were obtained from Wikidata (Vrandečić and Krötzsch, 2014). We transformed the entity alignment into a KGC task by augmenting the knowledge graph with additional edges representing the linguistic relations between the entity pairs.

Model	Hits@1	Hits@3	Hits@10
Prix-LM	19.04	28.58	37.49
Ours	**21.39**	**31.99**	**43.50**
w/o local	21.11	31.64	42.23
w/o global	19.45	29.51	42.71
w/o mask	20.61	30.12	42.11
Ours+RotatE	21.15	31.43	43.32
Ours+ComplEx	19.71	30.52	41.87
Ours+GAN	20.71	31.36	43.14
Ours+DV	18.98	29.61	41.60

Table 4: This table presents the average results of seven languages of the ablation experiment which investigates the impact of different methods for acquiring global and local knowledge and different module effects. The results of the experiment provide insights into the relative importance of these different methods for improving the performance of the mKGC.

meaningful relationships between entities. In contrast, the use of PLMs, as employed in our method, can effectively address the issue of data sparsity and still achieve notable impact on performance. Overall, these results demonstrate the effectiveness of our approach in comparison to the use of PLMs alone for mKGC.

4.4 Cross-lingual Entity Alignment

To assess the generalizability of the proposed method, we conduct a comparison on the entity alignment task. As shown in Table 3, we compared the proposed method with the Prix-LM. This comparison allowed us to assess the performance of the proposed method on a different task and determine its potential for use in a wider range of applications. The results show our method The results of the comparison indicate that our proposed method outperforms the Prix-LM in most of the evaluation indicators. This suggests that our method is able to generalize well to different tasks and is capable of achieving improved performance on the entity alignment task. Counterintuitively, the results show that languages with fewer resources tend to yield better performance. This may be due to the fact that the relationship between low-resource entity pairs is relatively simple and easier for the model to learn.

4.5 Ablation Experiment

Our proposed method introduces a novel approach for extracting both global and local knowledge through the use of a scoring function function and the maximization of mutual information. As shown in Table 4, we conducted an extensive comparison with various alternatives for scoring function and mutual information estimation, showcasing the superior performance of the proposed method. And

we also verified the effect of different module, our findings indicate that the use of global features is better than local features, and that the difference between local and global results on the H10 metric is minimal. This supports our expectation that local features improve the ranking of entity types. Using the tasks matrix reduces some of the noise and allows faster convergence, which is the key to improving performance. We will include these results in our revised manuscript and provide a detailed discussion on their implications.

4.6 Answer Length Comparison

As shown in Figure 5, we compared the performance of the proposed method on answers of different lengths to assess its robustness. The results of this comparison demonstrate that the proposed method exhibits strong performance across a range of answer lengths, indicating its ability to handle diverse inputs effectively. The results show that our method outperforms the baseline in terms of Hits values for answers of various lengths, with particularly strong performance on short answers.

4.7 Case Study

As shown in Figure 4, we compare the performance of our method with Prix-LM on a set of real examples. The predicted answers generated by both methods are presented and analyzed in order to evaluate the effectiveness of each approach for mKGC. The results of these case studies provide additional evidence for the effectiveness of our approach in comparison to the baseline model. Our analysis of the top three cases reveals that our method produces a higher number of predictions of the same type as the correct answer compared to

Language	Query	Answer	Prediction by Prix-LM	Predicition by Ours
Deutsch	(Napa Valley AVA, Sortenreine Weine, ?)	Merlot	Nunavut | Chardon | Nuclear Death | Cabernet Blanc | Çaycuma | Nuclear Blast | Symphony | Symphonic Metal | Porter (Band) | Nuclear Death Terror	**Merlot** | Grauburgunder | Cabernet Blanc | Chardonel | Trousseau Gris | Sauvignon Gris | Cabernet Jura | Chardon | Cabernet Cubin | Cabernet Cortis
Suomi	(Martin Thörnberg, syntymäpaikka, ?)	Ruotsi	Helsinki | Hamburg | Bern | Ullevi | Fribourg | Kołobrzeg | Baden-Württemberg | Weimar | Lahti | Kopparberg	**Ruotsi** | Suomi | Englanti | Yhdysvallat | Helsinki | Espoo | Häme | Kotka | Lahti | Vantaa
French	(Région wallonne, langue, ?)	Allemand	Malte | Hollande | Langue française | Belgique | Malé | Portugais | Budu | Langue anglaise | Suédois | Langue nationale	**Allemand** | Langue française | Anglais | Hongrois | Italien | Hollandais | Française | Arménien | Slavon d'église
Italiano	(One Piece:Unlimited World Red, piattaforma, ?)	PlayStation Vita	PlayStation 2 | PlayStation 4 | Game Boy Advance | N-Gage | PlayStation | Playstation 3 | PlayStation Vita | Xbox 360 | Nintendo DS | IOS	**PlayStation Vita** | PlayStation 2 | PlayStation | PlayStation 4 | Xbox | PlayStation Portable | Playstation 2 | Xbox 360 | Microsoft Windows | Playstation 3
Magyar	(Joszif Sztálin, halálozási hely, ?)	Szovjetunió	Luxembourg | Celje | Athén | Roma | Braunschweig | Szécsény | Rostock | Molde | Ħamrun | Rodolfo	**Szovjetunió** | Jeruzsálem | Lengyelország | Kolozsvár | Temesvár | Kairó | Szentgotthárd | Szarajevó | Szentpétervár | Szaloniki
Turkish	(Gérson Caçapa, doğumyeri, ?)	Brezilya	João Pessoa | New York | La Paz | Estonya | Portekiz | João Paulo | Rio de Janeiro | João Havelange | Letonya | Fukuoka;	**Brezilya** | Portekiz | Rio de Janeiro | Minas Gerais | São Luís | Arjantin | Porto | Salvador | Fortaleza | Porto Alegre

Figure 4: This figure presents a comparison of the performance of our method and baseline model on a set of case studies. The **blue** font is used to indicate that the predicted answer aligns with the golden answer type. The **bold** font in the predicted answer signifies the correct answer.

the baseline model. This finding suggests that our approach effectively addresses the task bias and demonstrates the adaptability of the PLM for the KGC task. Despite, the predicted answer types in the bottom three examples are all same, our method is able to accurately identify the correct answer. This demonstrates the robustness and effectiveness of our approach in generating accurate results even in situations where the predicted answers type are similar.

5 Related Work

5.1 Embedding-based Methods for KGC

There has been a amount of research focused on developing embedding-based methods for finding potential knowledge within a knowledge graph (Wang et al., 2017; Dai et al., 2020). These methods typically involve representing entities and relations within the graph as low-dimensional vector embeddings. Such like TransE (Bordes et al., 2013) makes entity and relation vectors follow the translational principle $\mathbf{h} + \mathbf{r} = \mathbf{t}$. The choice of scoring function and the specific vector space used can have a significant impact on the performance of the method, including RotatE(Sun et al., 2019), TransH (Wang et al., 2014), HolE(Nickel et al., 2016), ComplEx(Trouillon et al., 2016). However embedding-based methods may not fully consider the background knowledge that is implicit in the text associated with entities and relations.

5.2 Pretrained Language Models for KGC

Recently, some research leverage pretrained language models to complete KGC task. There methods represent entities and relations by PLMs, and score high for positive triplets (Lv et al., 2022; Kim et al., 2020). This manner enables the introduction of knowledge that has already been learned in PLMs.
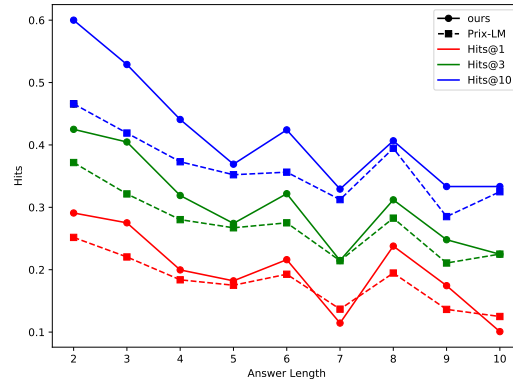

Figure 5: The figure presents the results of the Hits@k evaluation metric for a mKGC task, focusing on answers of varying lengths. In order to facilitate a more straightforward analysis, the results are limited to those sets of lengths that have more than 100 occurrences.

To fully utilize the PLM, some research focus on generative paradigm for knowledge graph construction (Ye et al., 2022). GenKGC (Xie et al., 2022) transforms knowledge graph completion into a sequence-to-sequence generation task base on pretrained language model and propose relation-guided demonstration and entity-aware hierarchical decoding. COMET (Bosselut et al., 2019) propose the Commonsense Transformer to generate commonsense automatically. KGT5 (Saxena et al.,

2022) consider KG link prediction as sequence-to-sequence tasks base on a single encoder-decoder Transformer. It reduce model size for KG link prediction compare with embedding-based methods. While previous efforts to utilize PLMs for KGC have demonstrated effectiveness, they have not fully considered the inherently knowledge-based nature of KGC tasks. This oversight may hinder the full potential of such models in addressing the unique challenges and requirements of KGC.

6 Conclusion

Our work improve the multilingual knowledge graph completion performance base on PLM and generative paradigms. We propose two two knowledgeable tasks to integrate global and local knowledge into answer generation given a query. The global knowledge improves the type consistency of the generated answers. Local knowledge enhances the accuracy of answer generation. We conducted experiments and the results showed that the proposed method is better than the previous model.

7 Limitations

While our approach effectively predicts the relationships between entities in a knowledge graph, there are limitations in the scope of knowledge graph resources that can be modeled. The knowledge graph contains a vast array of resources, including attributes, descriptions, and images, which are not easily captured by embedding-based methods, but can be effectively modeled using PLMs. To improve the compatibility of KGC with actual needs, it is necessary to consider a broader range of data types in the knowledge graph and develop complementary methods to effectively incorporate them.

8 Ethics Statement

This paper proposes a method for Multilingual Knowledge Graph Completion, and the experiments are conducted on public available datasets. As a result, there is no data privacy concern. Meanwhile, this paper does not involve human annotations, and there are no related ethicalconcerns.

9 Acknowledgements

References

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. 2018. Mutual information neural estimation. In *International conference on machine learning*, pages 531–540. PMLR.

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

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

Zhe Chen, Yuehan Wang, Bin Zhao, Jing Cheng, Xin Zhao, and Zongtao Duan. 2020. Knowledge graph completion: A review. *Ieee Access*, 8:192435–192456.

Yuanfei Dai, Shiping Wang, Neal N Xiong, and Wenzhong Guo. 2020. A survey on knowledge graph embedding: Approaches, applications and benchmarks. *Electronics*, 9(5):750.

Jinhua Du, Yan Huang, and Karo Moilanen. 2021. Knowledge-aware leap-lstm: Integrating prior knowledge into leap-lstm towards faster long text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12768–12775.

Yuwei Fang, Shuohang Wang, Yichong Xu, Ruochen Xu, Siqi Sun, Chenguang Zhu, and Michael Zeng. 2022. Leveraging knowledge in multilingual commonsense reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3237–3246.

Xu Han, Shulin Cao, Xin Lv, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. 2018. Openke: An open toolkit for knowledge embedding. In *Proceedings of the 2018 conference on empirical methods in natural language processing: system demonstrations*, pages 139–144.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

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.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195.

Junyi Li, Tianyi Tang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. Pretrained language models for text generation: A survey. *arXiv preprint arXiv:2105.10311*.

Bill Yuchen Lin, Seyeon Lee, Xiaoyang Qiao, and Xiang Ren. 2021a. Common sense beyond english: Evaluating and improving multilingual language models for commonsense reasoning. *arXiv preprint arXiv:2106.06937*.

Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2021b. Pretrained transformers for text ranking: Bert and beyond. *Synthesis Lectures on Human Language Technologies*, 14(4):1–325.

Xin Lv, Yankai Lin, Yixin Cao, Lei Hou, Juanzi Li, Zhiyuan Liu, Peng Li, and Jie Zhou. 2022. Do pretrained models benefit knowledge graph completion? a reliable evaluation and a reasonable approach. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3570–3581.

Christian Meilicke, Melisachew Wudage Chekol, Daniel Ruffinelli, and Heiner Stuckenschmidt. 2019. Anytime bottom-up rule learning for knowledge graph completion. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 3137–3143.

Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. 2016. Holographic embeddings of knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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 2463–2473.

Apoorv Saxena, Adrian Kochsiek, and Rainer Gemulla. 2022. Sequence-to-sequence knowledge graph completion and question answering. *arXiv preprint arXiv:2203.10321*.

Sanket Shah, Anand Mishra, Naganand Yadati, and Partha Pratim Talukdar. 2019. Kvqa: Knowledge-aware visual question answering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 8876–8884.

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*.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International conference on machine learning*, pages 2071–2080. PMLR.

Michael Tschannen, Josip Djolonga, Paul K Rubenstein, Sylvain Gelly, and Mario Lucic. 2019. On mutual information maximization for representation learning. In *International Conference on Learning Representations*.

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

Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. 2019. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 968–977.

Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI conference on artificial intelligence*, volume 28.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Xin Xie, Ningyu Zhang, Zhoubo Li, Shumin Deng, Hui Chen, Feiyu Xiong, Mosha Chen, and Huajun Chen. 2022. From discrimination to generation: Knowledge graph completion with generative transformer. *arXiv preprint arXiv:2202.02113*.

Hongbin Ye, Ningyu Zhang, Hui Chen, and Huajun Chen. 2022. Generative knowledge graph construction: A review. *arXiv preprint arXiv:2210.12714*.

Yan Zhang, Ruidan He, Zuozhu Liu, Kwan Hui Lim, and Lidong Bing. 2020. An unsupervised sentence embedding method by mutual information maximization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1601–1610.

Wenxuan Zhou, Fangyu Liu, Ivan Vulić, Nigel Collier, and Muhao Chen. 2022. Prix-LM: Pretraining for multilingual knowledge base construction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5412–5424, Dublin, Ireland. Association for Computational Linguistics.

Yucheng Zhou, Xiubo Geng, Tao Shen, Wenqiang Zhang, and Daxin Jiang. 2021. Improving zero-shot cross-lingual transfer for multilingual question answering over knowledge graph. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5822–5834.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 7

☒ A2. Did you discuss any potential risks of your work?
Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☐ Did you use or create scientific artifacts?

Not applicable. Left blank.

☐ B1. Did you cite the creators of artifacts you used?
Not applicable. Left blank.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Not applicable. Left blank.

C ☒ Did you run computational experiments?

Left blank.

☐ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Not applicable. Left blank.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
4.2

☒ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Left blank.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
4.2

D   ☒   Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.