Does Pre-trained Language Model Actually Infer Unseen Links in Knowledge Graph Completion?

Anonymous ACL submission

Abstract

Knowledge graphs (KGs) consist of links that describe relationships between entities. Due to the difficulty of manually enumerating all relationships between entities, automatically completing them is essential for KGs. Knowledge Graph Completion (KGC) is a task that infers unseen relationships between entities in a KG. Traditional embedding-based KGC methods (e.g. RESCAL, TransE, DistMult, ComplEx, RotatE, HAKE, HousE, etc.) infer missing links using only the knowledge from training data. In contrast, the recent Pre-trained Language Model (PLM)-based KGC utilizes knowledge obtained during pre-training, which means it can estimate missing links between entities by reusing memorized knowledge from pre-training without inference. This part is problematic because building KGC models aims to infer unseen links between entities. However, conventional evaluations in KGC do not consider inference and memorization abilities separately. Thus, a PLM-based KGC method, which achieves high performance in current KGC evaluations, may be ineffective in practical applications. To address this issue, we analyze whether PLM-based KGC methods make inferences or merely access memorized knowledge. For this purpose, we propose a method for constructing synthetic datasets specified in this analysis and conclude that PLMs acquire the inference abilities required for KGC through pre-training, even though the performance improvements mostly come from textual information of entities and relations.

1 Introduction

A knowledge graph (KG) is graph-structured data that includes relationships between entities as links. KGs are useful resources to inject external knowledge into NLP models. Since manually considering all possible links between entities is difficult, it is important to use a task such as KG completion (KGC), which automatically completes unseen links from seen ones in a KG.

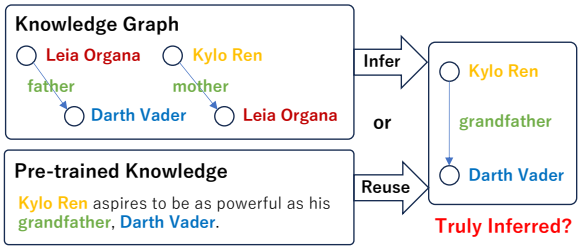

Figure 1: PLM-based KGC can reuse pre-trained knowledge of unseen links instead of inferring them.

As a basic method for KGC, KG embedding (KGE) is a popular chioce for this task. KGE embeds entities and their relationships as continuous vectors and then calculates the plausibility of unseen links. Traditional KGE methods learn these embeddings only from a target KG (Nickel et al., 2011; Bordes et al., 2013; Yang et al., 2015; Trouillon et al., 2016; Sun et al., 2019; Zhang et al., 2020; Li et al., 2022). Thus, they purely infer unseen links to complete KGs.

Similar to other NLP fields, KGC also utilizes pre-trained language models (PLMs) (Yao et al., 2019; Lv et al., 2022; Shen et al., 2022; Zhang et al., 2022; Choi et al., 2021; Choi and Ko, 2023; Wang et al., 2021a,c, 2022; Xie et al., 2022; Saxena et al., 2022; Chen et al., 2022; Xie et al., 2023; Zhu et al., 2023). Unlike traditional KGE methods, PLM-based KGE methods can access knowledge obtained through pre-training. This characteristic makes PLM-based KGE methods achieve higher KGC performance than the traditional KGE methods.

However, since the purpose of KGC is to infer unseen links from seen links in KGs, we should separately consider the performance gain from reusing the information of the unseen links obtained in pre-training and inferring unseen links from the seen links in KGs. Figure 1 shows an example of PLM-based KGC. As we can see, PLM-based

Available Information	Traditional KGC	Pretrained Language Model (PLM)-based KGC									
		BASE		VIRTUAL WORLD		ANONYMIZED ENTITIES		INCONSISTENT DESCRIPTIONS		FULLY ANONYMIZED	
		Pre.	Rand.	Pre.	Rand.	Pre.	Rand.	Pre.	Rand.	Pre.	Rand.
Seen links in a KG	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Descriptions of entities/relations	✗	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗
Pre-trained knowledge of KGs	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
Abilities obtained by pre-training	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗

Table 1: Available information for each configuration. When compared, we can reveal what improves the KGC performance on PLMs. BASE denotes the setting on the original data, and VIRTUAL WORLD (§3.1), ANONYMIZED ENTITIES (§3.2), INCONSISTENT DESCRIPTIONS (§3.3), and FULLY ANONYMIZED (§3.4) denote the settings on our synthetic datasets. Pre. and Rand. denote the setting with pre-trained and randomly initialized weights, respectively.

KGC methods can estimate unseen links without inferring them from seen links in the target KG. This characteristic is problematic because we cannot estimate the inference ability of PLM-based KGC methods for truly unseen relationships between entities in KGs.

To address this issue, we propose a method to create synthetic datasets for KGC tasks intended to separately evaluate KGC performance by reusing the knowledge from pre-training corresponding to target unseen links and inferring from seen links in KGs. More specifically, we change the textual information of entities and relations while maintaining the graph structure of KGs, thereby creating an environment different from the PLMs' knowledge corresponding to unseen links in KGs. Due to this change, PLMs cannot rely on their pre-trained knowledge and must rely on their pure inference abilities. Table 1 summarizes the configurations provided by our synthetic datasets. By comparing these configurations, we can reveal what actually contributes to the KGC performance of PLMs.

We conducted experiments on various pre-trained models under our controlled synthetic dataset constructed from WN18RR (Dettmers et al., 2018), FB15k-237 (Toutanova and Chen, 2015), and Wikidata5m (Wang et al., 2021c). The results showed that PLMs acquire the inference abilities required for KGC in pre-training but rely more on textual information of entities and relations in KGs. We also observed that the KGC performance of PLM-based KGC without pre-trained information is comparable to or lower than that of TransE, the traditional KGC. This finding indicates the importance of both traditional and PLM-based KGC methods.

2 Knowledge Graph Completion

2.1 Task Definition for KGs with Descriptions

We assume that a KG $\mathcal{G}$ includes descriptions defined as a tuple, $\mathcal{G}=(\mathcal{E},\mathcal{R},\mathcal{T},\mathcal{D})$, where $\mathcal{E}$ denotes a set of entities, $\mathcal{R}$ denotes a set of relations, $\mathcal{T}$ denotes a set of triples, and $\mathcal{D}$ denotes descriptions for the entities. Each triple is represented as $(h,r,t)\in\mathcal{T}$, where h and $t\in\mathcal{E}$ are the head and tail entities, respectively, and $r\in\mathcal{R}$ is the relation. Every entity $e_i\in\mathcal{E}$ has a corresponding description $d_i\in\mathcal{D}$. KGC is a task to fill in the missing triples in KGs. Specifically, this involves using a query, a partial triple $(h,r,?)$ or $(?,r,t)$ to predict its answer, an entity at the position of $?$, within the KG. Note that the prediction is exclusively focused on entities; predicting their corresponding descriptions is not required.

KGC is often evaluated by rank prediction metrics such as Hits@k ($k\in\{1,3,10\}$), mean rank (MR), and mean reciprocal rank (MRR). Hits@k calculates the proportion of correct entities ranked among the top-k, MR is the average rank of all test triples, and MRR is the average reciprocal rank of all test triples.

2.2 KGC Methods

Traditional KGC methods, e.g., RESCAL (Nickel et al., 2011), TransE (Bordes et al., 2013), DistMult (Yang et al., 2015), ComplEx (Trouillon et al., 2016), RotatE (Sun et al., 2019), HAKE (Zhang et al., 2020), and HousE (Li et al., 2022), primarily focus on the structure of KGs, without considering the extensive textual information.

However, recent advancements integrating PLMs have allowed KGC methods to encode text (Yao et al., 2019; Lv et al., 2022; Shen et al., 2022; Zhang et al., 2022; Choi et al., 2021; Choi

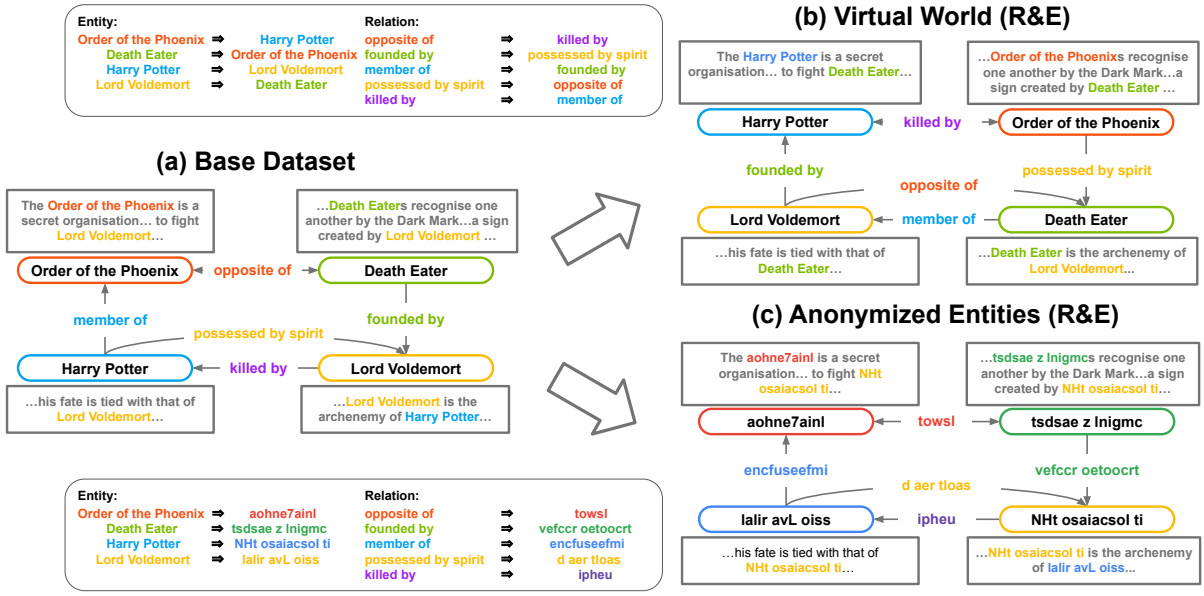

Figure 2: (a): Example of a KG with entity descriptions for PLM-based methods. Each entity has a corresponding description. (b) and (c) are the datasets used in this study. We primarily apply two methods for creating these datasets in VIRTUAL WORLD (§3.1) and ANONIMIZED ENTITIES (§3.2). (b) described in VIRTUAL WORLD (§3.1) involves swapping the names assigned to entities and relations in the base dataset respectively. (c) described in ANONIMIZED ENTITIES (§3.2) substitutes the names of entities and relations in the base dataset with random strings. Note that in both procedures, any entities appearing within the description text are replaced with their corresponding transformed names to maintain the graph structure within the descriptions.

and Ko, 2023; Wang et al., 2021a,c, 2022) or generate facts (Xie et al., 2022; Saxena et al., 2022; Chen et al., 2022; Xie et al., 2023; Zhu et al., 2023), thereby enhancing the KGC performance. These methods can be broadly divided into two categories based on their usage: discrimination-based methods that utilize PLM encoders, and generation-based methods that utilize PLM decoders (Pan et al., 2023) (see Appendix A for the details).

3 Synthetic Dataset Construction

To analyze the behavior of PLM-based KGC methods, we create synthetic data corresponding to each setting in Table 1. These settings affect the usable information of the PLM-based KGC methods but do not influence the traditional KGE methods. We explain the details for each setting in the following subsections.

3.1 Virtual World

To separate the pre-trained knowledge of PLMs and a target KG, we create a virtual world by shuffling each entity and/or relation name in the KG.

As shown in Figure 2(b), we shuffle the textual information associated with each entity and/or relation while keeping the graph structure within the created synthetic dataset. To ensure there are no unshuffled elements, we shuffle the entities using the derangement algorithm by Martínez et al. (2008).

However, there are dramatically fewer relations compared to entities (e.g., ten relations for ten thousand entities), and if relations are shuffled, the triple remains unchanged in many cases.[1] To address

Algorithm 1: Derangement by Bipartite Graph

Data: Input array arr of size n, Set of removed edges $removed\_edges$
Result: Generated array res

Create an empty graph G;
for $i \leftarrow 0$ **to** $n-1$ **do**
  for $j \leftarrow 0$ **to** $n-1$ **do**
    if $arr[i] \neq arr[j]$ *and* $(arr[i], arr[j])$ *is not in* $removed\_edges$ **then**
      add edge $(i, n+j)$ in G;
    end
  end
end
$match \leftarrow$ maximum matching(G)
$res \leftarrow$ an empty list of size n;
for $i \leftarrow 0$ **to** $n-1$ **do**
  $index \leftarrow match[i] - n$;
  $res[i] \leftarrow arr[index]$;
end
return res

[1] In the case of (Johann Bernoulli, wasBornIn, Basel) and

these cases, we apply a derangement based on a bipartite graph (Iradmusa and Praeger, 2019; Horsley et al., 2020) in Algorithm 1 for relations.

In Algorithm 1, we introduce $removed\_edges$, a set to the bipartite graph-based derangement. Lines 4–6 in Algorithm 1 delete edges leading to multiple relations in a triplet $(h,*,t)$, thereby preventing transitions to these relations.[2] We use the Hopcroft-Karp algorithm (Hopcroft and Karp, 1971) for maximum bipartite matching.

Additionally, we use Trie search (Yata, 2013) to comprehensively search for entity representations within each description in Figure 2 and change them into their post-shuffled text representations. This procedure treats the relationships between entities within the descriptions while maintaining their original graph structure in the descriptions.

3.2 Anonymized Entities

VIRTUAL WORLD can separate the pre-trained knowledge of PLMs and a target KG. However, this setting may underestimate the KGC performance caused by the overwrap of the entity and/or relation names between pre-trained knowledge and the target KG.

The ANONYMIZED ENTITIES setting can solve this problem by replacing the textual information associated with each entity and/or relation with a random string while keeping the original graph structure within the dataset, as in Figure2(c). Afterward, we also replace the entity representations within the description with these random strings using Trie search, the same as VIRTUAL WORLD (§3.1).

Since the random strings should follow language characteristics, we first construct character-level unigram language models $P(s_i)$, including space characters from the set of textual information of each entity and relation.

Next, we generate random strings $\boldsymbol{s}=s_1,s_2,...,s_n$ based on the character-level unigram language model $p(\boldsymbol{s})$, i.e., the product of the probabilities of unigram character in the strings:

$$p(\boldsymbol{s})=\prod_{i=1}^{n} p(s_i). \quad (1)$$

We stop the generation of strings when an end-of-sequence symbol is sampled. The strings are treated as a series of independent characters, allowing us to generate entirely random strings without using information about co-occurrence between characters. However, we preserve information for the randomly sampled sequences across the entire dataset so that each entity or relation is replaced with a unique sequence avoiding duplicates.

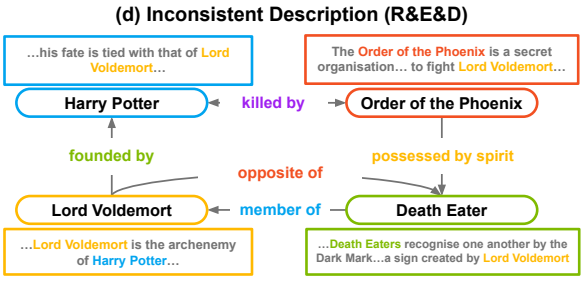

Figure 3: Example of a synthetic dataset created in INCONSISTENT DESCRIPTIONS (§3.3). Compared to Figure2(b) which shows an example of VIRTUAL WORLD (§3.1), the descriptions here also move to the same positions as the entities. Also, the entities in the descriptions do not change. At first glance, it appears the description explains the real-world relationships of the corresponding entities, but the relationships between entities within the synthetic dataset are actually broken.

3.3 Inconsistent Descriptions

To measure the effect of descriptions on PLM-based KGC, we isolate the entity and relation knowledge from the description by breaking the consistency between the graph structure and descriptions in addition to the shuffle of entity and/or relation names.

INCONSISTENT DESCRIPTIONS has two variations, one in which only the descriptions are shuffled and the other in which both the descriptions and entities/relations are shuffled. In the first variation, we derive the scenario in which there is no correspondence between an entity and its description by shuffling the set of descriptions via a derangement to get a new set $d'\in D'$. Then, we assign for each entity the new descriptions from D', i.e., $\forall e_i\in E, e_i{:}d_i\rightarrow d'_i$.

The second variation considers the descriptions and entities presented in Figure 3. The difference from Figure 2(b) for VIRTUAL WORLD (§3.1) lies in the way it handles the descriptions. In INCONSISTENT DESCRIPTIONS, descriptions are also shuffled together with the corresponding textual information when performing VIRTUAL WORLD, but the entities in the descriptions are preserved. In other words, when we map from e_i to e_j, we similarly map from d_i to d_j.

(Johann Bernoulli, diedIn, Basel), the swapping of the relations wasBornIn and diedIn does not change the triples.

[2] If $removed\_edges$ is empty, it is a normal derangement.

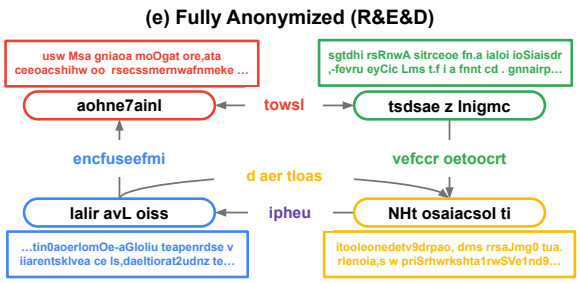

Figure 4: Example of a synthetic dataset created in FULLY ANONIMIZED (§3.4). Compared to Figure 2(c), which shows an example of ANONYMIZED ENTITIES (§3.2), the descriptions are here also changed into random strings. The descriptions become noisy information, and it becomes impossible to utilize any information from them.

Even though the descriptions explain the entities in the real world, they diverge from the relationships among entities in the dataset after the shuffle operation. Thus, if the model relies too much on the descriptions, it will be confused by this inconsistency.

3.4 Fully Anonymized

Figure 4 shows an example of FULLY ANONYMIZED, which is similar to ANONYMIZED ENTITIES (§3.2) in Figure 2(c) but differs in whether or not there is an operation on the descriptions. We replace the descriptions with random strings using the character-level unigram model utilized in ANONYMIZED ENTITIES (§3.2), while we keep the original structure of the KGs. This setting aims to mitigate underestimating the KGC performance caused by the overlap of the entity and/or relation names between pre-trained knowledge and the target KG. Note that the random string generation is applied independently to entities, relations, and descriptions. The key difference between FULLY ANONYMIZED and INCONSISTENT DESCRIPTIONS (§3.3) lies in whether the descriptions are readable sentences or not; if they are not, the PLMs in FULLY ANONYMIZED cannot rely on any pre-trained knowledge.

4 Experiments

4.1 Settings

Metrics We analyze how the inference capabilities are affected by each synthetic dataset (§3) measured with the Hits@10 metric on the test dataset and the validation dataset in the KGC task.[3]

Dataset	#entity	#relation	#train	#valid	#test
WN18RR	40,943	11	86,835	3,034	3,134
FB15k-237	14,541	237	272,115	17,535	20,466
Wikidata5m	4,594,485	822	20,614,279	5,163	6,894

Table 2: Dataset statistics.

Datasets We used WN18RR, FB15k-237, and Wikidata5m[4] as the base datasets; the details are shown in Table 2.[5] We applied VIRTUAL WORLD (§3.1) and ANONYMIZED ENTITIES (§3.2) to the entities and/or relations for creating synthetic datasets, resulting in a total of six types of datasets. Furthermore, we applied INCONSISTENT DESCRIPTIONS (§3.3) with and without VIRTUAL WORLD for entities and/or relations. INCONSISTENT DESCRIPTIONS (§3.4) is also applied with and without ANONYMIZED ENTITIES, and thus, we obtained additional six types of datasets. In total, we have 13 types of datasets, including the original one for each base dataset.

Comparison Methods We employ SimKGC (Wang et al., 2022) and kNN-KGE (Zhang et al., 2022) as Discriminative-based methods, and KGT5 (Saxena et al., 2022) and GenKGC (Xie et al., 2022) as Generation-based methods. We use the LambdaKG framework (Xie et al., 2023) as the base implementation, with hyper-parameters set to their default values. The seed value is fixed for all experiments.[6] We set early stopping for WN18RR and FB15k-237 when the Hits@10 value on the validation data did not improve for four epochs. For Wikidata5m, we conducted training only one epoch.[7] We also compare two cases: using pre-trained weights and setting weights randomly.

[3] We also measured Hits@1, Hits@3, MRR, and MR, and all showed similar trends. In this paper, we present the results using hits@10 for brevity.

[4] We follow the transductive setting in Wang et al. (2021c).

[5] We use the datasets with textual information provided by Yao et al. (2019) for WN18RR, FB15k-237, and by Wang et al. (2021c) for Wikidata5m.

[6] We conducted pilot studies with various seeds for several datasets and models. The variance observed was around 0.02, so a fixed seed value was chosen. For example, the Hits@10 scores in kNN-KGE on WN18RR applied with FULLY ANONYMIZED (§3.4) to all descriptions, entities, and relations were 0.426 ± 0.001 with three different seeds.

[7] We only report the results from SimKGC, as kNN-KGE could not be executed due to computational resource limitations, and both KGT5 and GenKGC did not produce scores under these settings. We conducted all experiments on a single NVIDIA A100 (40GB) or a single NVIDIA A6000 (48GB).

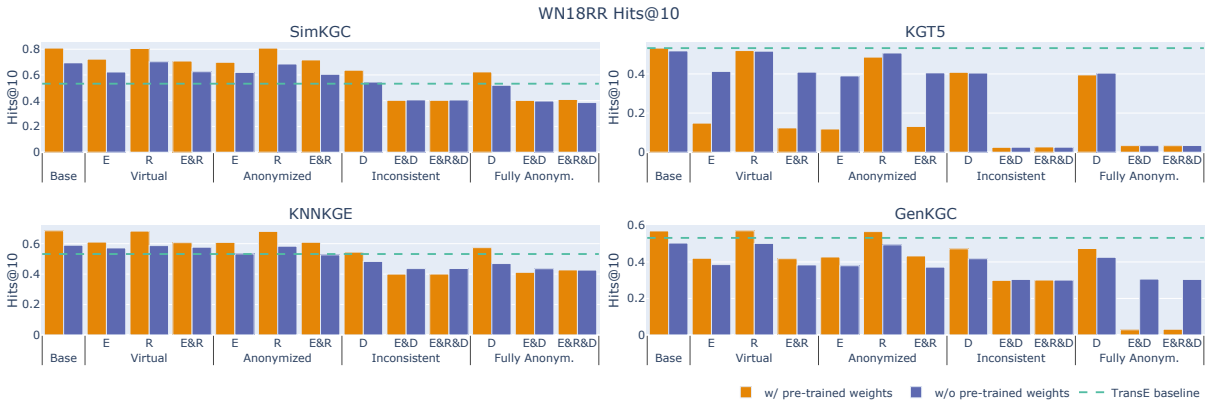

Figure 5: The hits@10 results on WN18RR. "E", "R", and "D" represent entity, relation, and description, respectively. For example, "E&R" denotes the application of the method to both entities and relations. For comparison, we have also included the hits@10 results on WN18RR by TransE reported by Nathani et al. (2019), which are the same score because the TransE model does not require textual information. The graphs on the left represent Discrimination-Based Methods, while those on the right represent Generation-Based Methods.

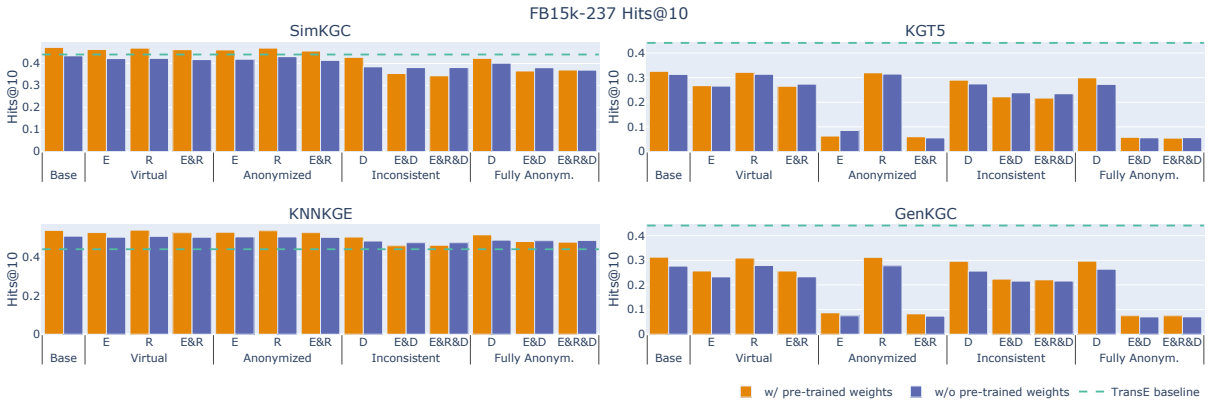

Figure 6: Hits@10 results on FB-15k-237. The supplementary explanation is the same as in Figure 5.

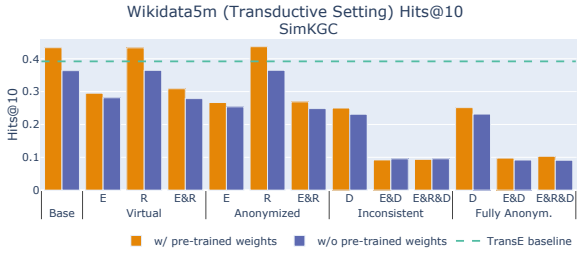

Figure 7: Hits@10 results on Wikidata5m by SimKGC. We have also included the Hits@10 results on WN18RR by TransE reported by Wang et al. (2021c).

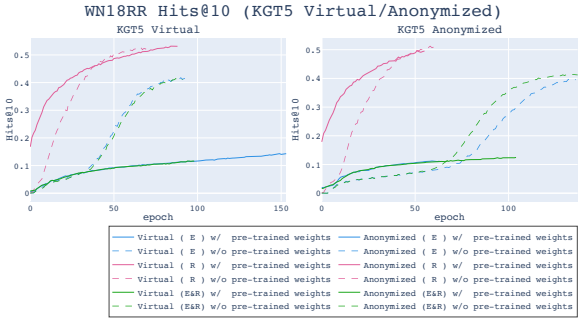

Figure 8: The plots show Hits@10 scores on WN18RR for the validation data at each epoch. The solid line represents using pre-trained weights, and the dashed line represents initializing weights randomly.

4.2 Results and Analysis

4.2.1 Effect of knowledge in PLMs

The results for each model and dataset on WN18RR and FB15k-237 are shown in Figures 5 and 6, and the results from SimKGC on Wikidata5m are shown in Figure 7. In the "Base" setting, all models with the pre-trained weights were better than those without them. When the models are trained without pre-training weights, they have to infer unseen links based only on information within the training

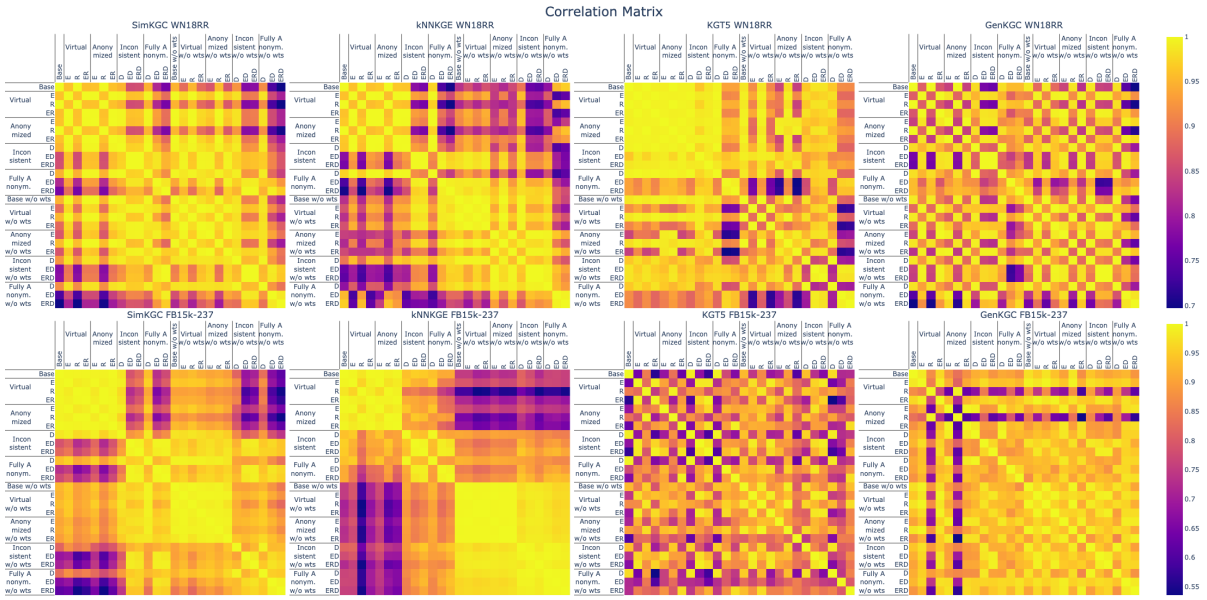

Figure 9: The correlation matrix (Pearson's correlation) shows the hits@10 values for the validation data for each dataset and each model."Virtual", "Anonymized", "Inconsistent", and "Fully Anonym." represent the methods applied in Sections 3.1, 3.2, 3.3, and 3.4, respectively. "E", "R", and "D" represent entity, relation, and description, respectively. For example, "ER" denotes the application of the method to both entities and relations. "w/o wts" means training from scratch with random initial values. The two graphs on the left are Discrimination-Based Methods, and the two on the right are Generation-based Methods.

#Relation	WN18RR (%)				FB15k-237 (%)				Wikidata5m (%)			
	Train	Valid	Test	Total	Train	Valid	Test	Total	Train	Valid	Test	Total
1	60.09	97.55	97.51	57.68	13.52	65.26	61.64	12.56	24.23	98.58	98.73	24.22
2	35.39	2.41	2.49	37.31	14.02	24.44	25.61	12.98	17.23	1.98	0.93	17.21
3	4.21	0.02	–	4.61	11.39	7.39	8.90	10.88	21.88	0.21	0.28	21.88
4	0.30	0.02	–	0.38	10.01	2.08	2.67	9.53	11.95	0.01	0.05	11.95
5	0.02	–	–	0.02	9.06	0.51	0.76	8.85	7.94	0.01	0.01	7.94
Over	–	–	–	–	42.00	0.31	0.41	45.19	16.78	–	–	16.79

Table 3: The number of relations assigned to each entity in each dataset. Note that some entities may be associated with multiple entities under certain entity and relation queries.

data of the KGC dataset.

Comparing "Base", "Virtual", and "Anonymized" settings, we can see performance degradations by restricted access to knowledge for entity names obtained in pre-training. However, the models without the pre-trained weights achieved better or at least comparable results, especially when changes were made to both entities and their descriptions, as you can see in the "Inconsistent" and "Fully Anonym." settings. From the result, we hypothesize that the performance gain by pre-trained weights in "Virtual" and "Anonymized" settings comes from the pre-trained ability to read textual information.

Figure 7 shows the importance of pre-trained knowledge for entity names in Wikidata5m. For the further analysis, we applied the interquartile range (IQR), an outlier detection method (Tukey, 1977), and the result show the significant performance gap between models with and without pre-trained weights only when entity names and their descriptions were unchanged. This finding indicates that PLM knowledge significantly contributes to the model's inference, especially in Wikidata5m.

4.2.2 Biases caused by PLM knowledge on inference for unseen links

We discussed the benefit of PLM knowledge in Section 4.2.1, but on the other hand, PLM knowledge may adversely affect the inference for unseen entities. Especially in Figures 5 and 6, it is clear that the difference between with and without pre-trained knowledge significantly affected the scores, particularly in the case of entity changes in KGT5.

Figure 8 shows the training curves of Hits@10 on WN18RR for the validation data. Remarkable

results were observed for the VIRTUAL WORLD and ANONYMIZED ENTITIES methods in KGT5: namely the models using pre-trained weights could not learn well, even with sufficient epochs of training, whereas the models without pre-trained weights exhibited inference capability for unknown entities. These results suggest that while PLM knowledge helps infer unseen links, it may prevent the learning of new relationships due to the relationships included in the PLM knowledge.

4.2.3 Which factors (entity, relation, description) affect inference ability?

Figure 9 shows the correlation matrix of Hits@10 scores on the validation data for each dataset and model. In Figures 5, 6, and 9, the results from the base dataset and changes to relations indicate strong correlations in the learning process and Hits@10 scores in the test data. Therefore, the model is not affected by changes to relations when inferring unseen links. As shown in Table 2, the number of relations is significantly smaller than that of entities. Moreover, Table 3 reveals entities with only one assigned relation in the KGC dataset: 12% in FB15k-237 and over 50% in WN18RR. This suggests that the models can infer connections between entities without considering their actual relations.

Figure 9 also shows a correlation between VIRTUAL WORLD and ANONYMIZED ENTITIES, indicating that which kind of textual information is used for inference is less important than than the consistency in relationships between entities in each triplet. Additionally, when changing both the entity and the description, the score decreases in Figures 5 and 6. Table 4 shows how many entities to predict are included in the description of query entities; in WN18RR, about 15 % of the entities may be able to solve the KGC task just by extracting information from the description. Changes to the description only are less likely to be affected, but changing both the entity and the description eliminates clues to the answer from both, leading to a decrease in the inference capabilities with PLM.

4.2.4 Effect of model structures on performance

When comparing Generation-based methods with Discrimination-based methods, the former are substantially affected by random strings of entities. As shown in Figure 5, KGT5 and GenKGC without the pre-trained weights learn better than those that have them. Moreover, Figure 8 shows that scores do not improve even with sufficient training, which suggests that the difference in scores is not due to the early stopping. Thus, PLM knowledge prevents learning new relationships from descriptions in Generation-based methods.

Moreover, Generation-based methods are influenced by the string of the output entity, as seen in Figures from 5 to 7. On the other hand, Discrimination-based methods are less affected by the textual information, in contrast to Generation-based methods that are affected by random strings that lack the characteristics of language and are thus unsuitable for generation (see Appendix B for further analysis).

	Train (%)	Valid (%)	Test (%)	Total (%)
WN18RR	15.03	15.62	15.34	15.06
FB15k-237	6.11	4.68	4.50	5.92
Wikidata5m	4.58	4.99	4.58	4.58

Table 4: Percentage of target entities to predict is included in the description of the query entity for each dataset. These triplets can be solved by simply extracting information from the descriptions without performing any inference in the KGC tasks.

5 Conclusion

In this study, we proposed a method for evaluating the inference ability of PLM-based KGC methods by separately considering the information related to unseen links in KGs. Using this method as a basis, we developed synthetic datasets that focused on the structure of KGs and changed only textual information, maintaining graph structure. Then, we compared PLM-based KGC methods using these datasets.

The comparison results show that PLMs acquire the inference abilities for KGC in pre-training, whereas in KGs, they rely more on the textual information of entities and relations. Further, we observed that the KGC performance of PLM-based KGC without pre-trained knowledge is comparable to or lower than that of TransE, the traditional KGC. This highlights the importance of using both traditional and PLM-based KGC methods. Please see Appendix C for more detailed information on improving the current KGC evaluation based on the insights from our work.

6 Limitations

In this study, we investigated the inference abilities of PLM-based KGC methods empirically, without focusing on theoretical verification. Furthermore, while our focus was on KGC, we did not verify whether these findings could be applied to other downstream tasks. Therefore, our future work will aim to generalize this empirical study and perform verification across various downstream tasks.

7 Ethical Considerations

In this study, we have created synthetic datasets derived from existing KG datasets that have cleared ethical issues following published conferences' policies. Therefore, our created datasets do not introduce any ethical problems.

References

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 2787–2795, Red Hook, NY, USA. Curran Associates Inc.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Stefan. Büttcher, Charles L. A Clarke, and Gordon V. Cormack. 2010. *Information retrieval : implementing and evaluating search engines*. MIT Press, Cambridge, Mass.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2023. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*.

Chen Chen, Yufei Wang, Bing Li, and Kwok-Yan Lam. 2022. Knowledge is flat: A Seq2Seq generative framework for various knowledge graph completion. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4005–4017, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

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.

Bonggeun Choi and Youngjoong Ko. 2023. Knowledge graph extension with a pre-trained language model via unified learning method. *Knowledge-Based Systems*, 262:110245.

Ameet Deshpande, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, and Karthik Narasimhan. 2023. Toxicity in chatgpt: Analyzing persona-assigned language models.

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 and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. 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.

Nouha Dziri, Ehsan Kamalloo, Sivan Milton, Osmar Zaiane, Mo Yu, Edoardo M Ponti, and Siva Reddy. 2022a. FaithDial: A Faithful Benchmark for Information-Seeking Dialogue. *Transactions of the Association for Computational Linguistics*, 10:1473–1490.

Nouha Dziri, Sivan Milton, Mo Yu, Osmar Zaiane, and Siva Reddy. 2022b. On the origin of hallucinations in conversational models: Is it the datasets or the models? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5271–5285, Seattle, United States. Association for Computational Linguistics.

Shangbin Feng, Chan Young Park, Yuhan Liu, and Yulia Tsvetkov. 2023. From pretraining data to language models to downstream tasks: Tracking the trails of political biases leading to unfair NLP models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11737–11762, Toronto, Canada. Association for Computational Linguistics.

Benjamin Heinzerling and Kentaro Inui. 2021. Language models as knowledge bases: On entity representations, storage capacity, and paraphrased queries. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1772–1791, Online. Association for Computational Linguistics.

John E. Hopcroft and Richard M. Karp. 1971. A $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM J. Comput.*, 2:225–231.

Daniel Horsley, Moharram Iradmusa, and Cheryl E Praeger. 2020. Generating Infinite Digraphs by Derangements. *The Quarterly Journal of Mathematics*, 72(3):961–974.

Moharram N. Iradmusa and Cheryl E. Praeger. 2019. Derangement action digraphs and graphs. *European Journal of Combinatorics*, 80:361–372. Special Issue in Memory of Michel Marie Deza.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12).

Maël Jullien, Marco Valentino, Hannah Frost, Paul O'regan, Donal Landers, and André Freitas. 2023. SemEval-2023 task 7: Multi-evidence natural language inference for clinical trial data. In *Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2216–2226, Toronto, Canada. Association for Computational Linguistics.

Masahiro Kaneko, Danushka Bollegala, and Naoaki Okazaki. 2022a. Debiasing isn't enough! – on the effectiveness of debiasing MLMs and their social biases in downstream tasks. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1299–1310, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Masahiro Kaneko, Aizhan Imankulova, Danushka Bollegala, and Naoaki Okazaki. 2022b. Gender bias in masked language models for multiple languages. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2740–2750, Seattle, United States. Association for Computational Linguistics.

Sayash Kapoor and Arvind Narayanan. 2022. Leakage and the reproducibility crisis in ml-based science.

Philippe Laban, Wojciech Kryściński, Divyansh Agarwal, Alexander R. Fabbri, Caiming Xiong, Shafiq Joty, and Chien-Sheng Wu. 2023. Llms as factual reasoners: Insights from existing benchmarks and beyond.

Faisal Ladhak, Esin Durmus, Mirac Suzgun, Tianyi Zhang, Dan Jurafsky, Kathleen McKeown, and Tatsunori Hashimoto. 2023. When do pre-training biases propagate to downstream tasks? a case study in text summarization. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3206–3219, Dubrovnik, Croatia. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Rui Li, Jianan Zhao, Chaozhuo Li, Di He, Yiqi Wang, Yuming Liu, Hao Sun, Senzhang Wang, Weiwei Deng, Yanming Shen, Xing Xie, and Qi Zhang. 2022. HousE: Knowledge graph embedding with householder parameterization. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 13209–13224. PMLR.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Xin Lv, Yankai Lin, Yixin Cao, Lei Hou, Juanzi Li, Zhiyuan Liu, Peng Li, and Jie Zhou. 2022. Do pre-trained 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, Dublin, Ireland. Association for Computational Linguistics.

Inbal Magar and Roy Schwartz. 2022. Data contamination: From memorization to exploitation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 157–165, Dublin, Ireland. Association for Computational Linguistics.

Conrado Martínez, Alois Panholzer, and Helmut Prodinger. 2008. *Generating Random Derangements*, pages 234–240.

Nick McKenna, Tianyi Li, Liang Cheng, Mohammad Javad Hosseini, Mark Johnson, and Mark Steedman. 2023. Sources of hallucination by large language models on inference tasks.

Nicholas Meade, Elinor Poole-Dayan, and Siva Reddy. 2022. An empirical survey of the effectiveness of debiasing techniques for pre-trained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1878–1898, Dublin, Ireland. Association for Computational Linguistics.

Inderjeet Nair and Natwar Modani. 2023. Exploiting language characteristics for legal domain-specific language model pretraining. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2516–2526, Dubrovnik, Croatia. Association for Computational Linguistics.

Deepak Nathani, Jatin Chauhan, Charu Sharma, and Manohar Kaul. 2019. Learning attention-based embeddings for relation prediction in knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4710–4723, Florence, Italy. Association for Computational Linguistics.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 809–816, Madison, WI, USA. Omnipress.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2023. Unifying large language models and knowledge graphs: A roadmap. https://doi.org/10.48550/arXiv.2306.08302.

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, Hong Kong, China. Association for Computational Linguistics.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is chatgpt a general-purpose natural language processing task solver?

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Oscar Sainz, Jon Ander Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. 2023. Nlp evaluation in trouble: On the need to measure llm data contamination for each benchmark.

Apoorv Saxena, Adrian Kochsiek, and Rainer Gemulla. 2022. Sequence-to-sequence knowledge graph completion and question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2814–2828, Dublin, Ireland. Association for Computational Linguistics.

Jianhao Shen, Chenguang Wang, Linyuan Gong, and Dawn Song. 2022. Joint language semantic and structure embedding for knowledge graph completion. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1965–1978, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 2071–2080. JMLR.org.

John W. Tukey. 1977. *Exploratory Data Analysis*. Addison-Wesley.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, volume 33, pages 12388–12401. Curran Associates, Inc.

Bo Wang, Tao Shen, Guodong Long, Tianyi Zhou, Ying Wang, and Yi Chang. 2021a. Structure-augmented text representation learning for efficient knowledge graph completion. In *Proceedings of the Web Conference 2021*, WWW '21, page 1737–1748, New York, NY, USA. Association for Computing Machinery.

Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. 2021b. Adversarial GLUE: A multi-task benchmark for robustness evaluation of language models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Jindong Wang, Xixu HU, Wenxin Hou, Hao Chen, Runkai Zheng, Yidong Wang, Linyi Yang, Wei Ye, Haojun Huang, Xiubo Geng, Binxing Jiao, Yue Zhang, and Xing Xie. 2023. On the robustness of chatGPT: An adversarial and out-of-distribution perspective. In *ICLR 2023 Workshop on Trustworthy and Reliable Large-Scale Machine Learning Models*.

Liang Wang, Wei Zhao, Zhuoyu Wei, and Jingming Liu. 2022. 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.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021c. KEPLER: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Transactions on Machine Learning Research*. Survey Certification.

Xin Xie, Zhoubo Li, Xiaohan Wang, Yuqi Zhu, Ningyu Zhang, Jintian Zhang, Siyuan Cheng, Bozhong Tian, Shumin Deng, Feiyu Xiong, and Huajun Chen. 2023. Lambdakg: A library for pre-trained language model-based knowledge graph embeddings.

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. In *Companion Proceedings of the Web Conference 2022*, WWW '22, page 162–165, New York, NY, USA. Association for Computing Machinery.

Bishan Yang, Wen tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. KG-BERT: BERT for knowledge graph completion. *CoRR*, abs/1909.03193.

Susumu Yata. 2013. Marisa: Matching algorithm with recursively implemented storage. Accessed: 26 Jul 2023; Last updated: 26 Jun 2020.

Jifan Yu, Xiaozhi Wang, Shangqing Tu, Shulin Cao, Daniel Zhang-Li, Xin Lv, Hao Peng, Zijun Yao, Xiaohan Zhang, Hanming Li, Chunyang Li, Zheyuan Zhang, Yushi Bai, Yantao Liu, Amy Xin, Nianyi Lin, Kaifeng Yun, Linlu Gong, Jianhui Chen, Zhili Wu, Yunjia Qi, Weikai Li, Yong Guan, Kaisheng Zeng, Ji Qi, Hailong Jin, Jinxin Liu, Yu Gu, Yuan Yao, Ning Ding, Lei Hou, Zhiyuan Liu, Bin Xu, Jie Tang, and Juanzi Li. 2023. Kola: Carefully benchmarking world knowledge of large language models.

Ningyu Zhang, Xin Xie, Xiang Chen, Shumin Deng, Chuanqi Tan, Fei Huang, Xu Cheng, and Huajun Chen. 2022. Reasoning through memorization: Nearest neighbor knowledge graph embeddings. *CoRR*, abs/2201.05575.

Zhanqiu Zhang, Jianyu Cai, Yongdong Zhang, and Jie Wang. 2020. Learning hierarchy-aware knowledge graph embeddings for link prediction. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 3065–3072. AAAI Press.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.

Pei Zhou, Rahul Khanna, Seyeon Lee, Bill Yuchen Lin, Daniel Ho, Jay Pujara, and Xiang Ren. 2021. RICA: Evaluating robust inference capabilities based on commonsense axioms. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7560–7579, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yuqi Zhu, Xiaohan Wang, Jing Chen, Shuofei Qiao, Yixin Ou, Yunzhi Yao, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2023. Llms for knowledge graph construction and reasoning: Recent capabilities and future opportunities.

A Details of PLM-based KGC Methods

A.1 Discrimination-based Methods

The early PLM-based KGC methods such as KG-BERT (Yao et al., 2019), utilize an encoder-only PLMs like BERT (Devlin et al., 2019) to encode triples. They perform binary classification to assess the plausibility of a given triplet. KG-BERT transforms a triple (h,r,t) as follows:

$$x=[\text{CLS}]\text{Text}_h[\text{SEP}]\text{Text}_r[\text{SEP}]\text{Text}_t[\text{SEP}], \quad (2)$$

where Text_n represents textual representations of n. The PLM takes x as input and conducts binary classification using the [CLS] token $e_{[\text{CLS}]}$ from the final hidden state. It calculates the plausibility of the triples, which is formulated as follows:

$$\text{Score}(h,r,t)=\text{Sigmoid}(\text{MLP}(e_{[\text{CLS}]})). \quad (3)$$

Zhang et al. (2022); Choi et al. (2021); Choi and Ko (2023) involve filling the missing part of a triple with a [MASK] token and predicting it. The input sequence x is represented as follows:

$$x=[\text{CLS}]\text{Text}_h[\text{SEP}]\text{Text}_r[\text{SEP}][\text{MASK}][\text{SEP}]. \quad (4)$$

Nonetheless, simply predicting the [MASK] token does not facilitate direct entity prediction. Consequently, it introduces special tokens into the vocabulary to represent the corresponding entities for prediction. In the case of kNN-KGE (Zhang et al., 2022), an initial learning process is undertaken when introducing these special tokens to establish the relationship between the special tokens and the entities.

The prompt shown in Equation (5) is used to mask the special tokens that represent each entity e_i. With all other parameters fixed, the masked entity e_i is predicted using cross-entropy loss. This approach optimizes the embeddings of these entities, which are initially set to random values.

$$x_i=[\text{CLS}]\text{ the description of }[\text{MASK}]\text{ is }d_i\ [\text{SEP}], \quad (5)$$

Afterwards, a sentence similar to Eq. (4) is fed into the model, which then fine-tunes the model to predict the masked entity, as formulated:

$$P(t|h,r)=P([\text{ MASK }]=\text{t}|x,\Theta), \quad (6)$$

where Θ denotes the parameters of the model.

Finally, SimKGC (Wang et al., 2022), the state-of-the-art method employs two encoders. SimKGC splits the triple (h,r,t) into a question (h,r) and its answer t and uses their respective PLMs to encode them into vector space, which can be expressed as:

$$x_{(h,r)}=[\text{CLS}]\ \text{Text}_h\ [\text{SEP}]\ \text{Text}_r\ [\text{SEP}]\ , \quad (7)$$

$$x_t=[\text{CLS}]\ \text{Text}_t\ [\text{SEP}]\ . \quad (8)$$

Then, the [CLS] tokens from the final hidden state are extracted, with the embedding of $x_{(h,r)}$ represented as $e_{(h,r)}$ and the embedding of x_t represented as e_t. The final plausibility of the triples is scored as follows:

$$\text{Score}((h,r),t)=\cos\left(e_{(h,r)},e_t\right). \quad (9)$$

Essentially, the introduced model originally employs the BERT-base model, but it can use variants of BERT such as RoBERTa (Liu et al., 2019).

A.2 Generation-based Methods

Recently, novel KGC-based methods have been introduced that utilize Encoder-Decoder models, e.g., GenKGC (Xie et al., 2022), KGT5 (Saxena et al., 2022), or Decoder-only Large Language Models (LLMs), e.g., LambdaKG (Xie et al., 2023), AutoKG (Zhu et al., 2023), to directly generate the tail entity t. Unlike traditional KGC methods and discrimination-based methods, which can only complete the KGs using a predefined set of entity candidates, these generation-based methods have the potential to predict unknown entities not included in the candidate list. This capability unlocks the ability to predict any and all entities in the KGs.

When predicting the missing triple $(h,t,?)$, the model converts $x_{(h,r)}$ into a prompt specific to the models, then it into the encoder and generates x_t.

While there is potential to predict any and all entities, in practice, certain restrictions are put in place to focus the prediction towards entities within the KGs. For example, GenKGC introduces an entity-aware hierarchical decoder to place constraints on x_t. Furthermore, KGT5 utilizes generation-based PLMs, pre-trained with text descriptions specifically for KG representation. Notably, this is done from scratch with random initialization, rather than leveraging pre-trained models, indicating the effectiveness of a tailored approach for each dataset.[8] Regarding the foundational models, GenKGC employs BART-base (Lewis et al., 2020), while KGT5 utilizes T5-small (Raffel et al., 2020).

[8]The authors mention that using pre-trained weights can improve accuracy in some cases (https://github.com/intfloat/SimKGC/issues/1). They also discuss the challenge of training models on small datasets (https://github.com/apoorvumang/kgt5/issues/4).

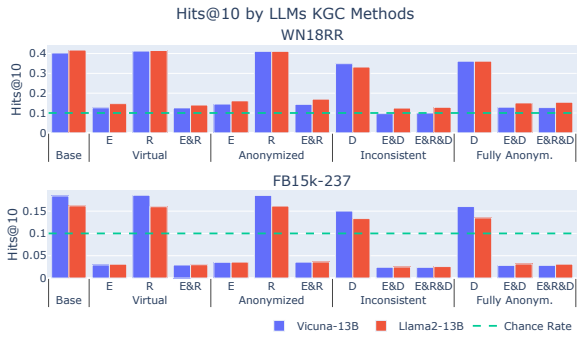

Figure 10: The results of Hits@10 using vicuna-13B and Llama2-13B in the LLMs KGC methods (Xie et al., 2023). The LLMs select 1 entity from selected 100 candidate entities by BM25. It generates 10 sentences, and it is checked whether the correct entity is included in these. The chance rate is 0.1 because it generated total 10 entities from 100 candidates.

Finally, some experimental KGC methods use decoder-only LLMs. These methods employ well-designed prompts to induce in-context learning. LambdaKG employs the information retrieval algorithm (BM25) (Büttcher et al., 2010) to construct prompts. It selects the top 100 most relevant entities from the dataset as potential answer candidates. Similarly, it retrieves the top 5 relevant triples as examples for few-shot learning. This information is aggregated into a single prompt, which is then used by LLMs to select and generate an answer. AutoKG addresses the KGC task in a 0-shot or 1-shot setting without employing an information retrieval algorithm. It treats the missing entity as a [MASK] token in the prompt and generates the corresponding value for the [MASK] token using LLMs.

B Inference capabilities under a zero-shot setting with LLMs

We evaluate the inference capabilities in a zero-shot setting by LLMs. We evaluate WN18RR and FB15k-237 using the LambdaKG method (Xie et al., 2023) described in Appendix A.1.[9] Figure 10 shows the results using Vicuna-13B (Zheng et al., 2023) and Llama2-13B (Touvron et al., 2023). The base dataset yields high hits@10 scores, but when entities are changed, the impact is high, and it is small when only descriptions are changed. However, LLMs don't know how the entity was changed, so the chance rate serves as an upper limit. Therefore, it is clear that inference by LLMs is based on pre-trained knowledge.

[9]Original LambdaKG uses GPT-3 (Brown et al., 2020), but we employ Vicuna-13B and Llama2-13B for reproducibility. These models have shown competitiveness to GPT-3 on the MT-bench Reasoning benchmark (Zheng et al., 2023). Furthermore, while the original setting calculates only Hits@1, this study calculates Hits@10 by considering the top 10 output probabilities.

C Exhortation to KGC

Datasets As discussed in Section 4.2.3, the information for relations has very little impact. Some entities are assigned only one relation, as shown in Table 3. Thus, if only the entity is known, it may be possible to infer the unknown entities without relation information. Traditional KGC methods without PLMs can learn the graph structure from scratch. In contrast, PLMs' knowledge can help with completion without relation information, as discussed in Section B. The current dataset focuses on entities, but it cannot accurately measure the effect of relations. Therefore, a dataset that specifically focuses on relations is needed.

Next, according to Table 4, it has become clear that the missing entity information is included in the descriptions of queries. Therefore, if we use descriptions in the KGC task, it can be considered a cheat setting, as it utilizes the information extraction capability from the text data in PLMs. The descriptions are indeed useful for disambiguation in entities, but they also provide too much information for inference, thus demonstrating information extraction capabilities. In the future, to measure the pure inference capabilities for unknown entities, descriptions should not be used in the KGC task for fair comparison.

Models As discussed in Section 4.2.1, PLMs' knowledge helps inferences for unknown entities. Therefore, when we evaluate filling in truly unknown links in KGs by KGC in the future, we should avoid using pre-trained weights. This suggests that PLM-based KGC methods with pre-trained weights create a cheat setting because they utilize external knowledge not included in datasets, which does not measure the pure inference capabilities for unknown entities in KGC tasks. It is essential to evaluate the model's performance based on the target KGC dataset only for a fair comparison.

D Related Work

Traditional KGC As introduced in §2.2, the traditional KGC methods, represented as RESCAL (Nickel et al., 2011), TransE (Bordes et al., 2013),

DistMult (Yang et al., 2015), ComplEx (Trouillon et al., 2016), RotatE (Sun et al., 2019), HAKE (Zhang et al., 2020), and HousE (Li et al., 2022) only focus on the structure of KGs, without considering the extensive textual information of KGs and pre-trained information. Thus, these models need to complete KGs only by their inference abilities.

PLM-based KGC As introduced in §2.2, PLM-based KGC methods encode text (Yao et al., 2019; Lv et al., 2022; Shen et al., 2022; Zhang et al., 2022; Choi et al., 2021; Choi and Ko, 2023; Wang et al., 2021a,c, 2022) or generate facts (Xie et al., 2022; Saxena et al., 2022; Chen et al., 2022; Xie et al., 2023; Zhu et al., 2023) based on pre-trained information to enhance KGC performance. There are two major categories, discrimination-based methods that utilize PLMs encoders and generation-based methods that utilize PLMs decoders (Pan et al., 2023). However, it is uncertain whether the performance improvement is actually caused by the enhanced ability of inference through pre-training or data leakage from pre-trained data. We aim to reveal that in our work.

Data Leakage in PLMs Some existing datasets for the downstream tasks are often directly mixed into the pre-training data (Magar and Schwartz, 2022; Kapoor and Narayanan, 2022; Sainz et al., 2023), and general PLMs are not able to answer questions correctly in downstream tasks that require domain-specific knowledge excluded from the pre-trained data (Wang et al., 2023; Jullien et al., 2023; Nair and Modani, 2023).

Inference Ability of PLMs Several studies (Zhou et al., 2021; Wang et al., 2021b; Zhu et al., 2023; Zheng et al., 2023; Yu et al., 2023; Laban et al., 2023; Qin et al., 2023) evaluate the inference abilities of PLMs, but they ignored the impact of the PLMs' memorization abilities in inference. Therefore, the inference abilities of PLMs remain unclear. While the memorization abilities of PLMs are beneficial (Petroni et al., 2019; Roberts et al., 2020; Heinzerling and Inui, 2021; Wei et al., 2022; Carlini et al., 2023), they can introduce bias (Vig et al., 2020; Kaneko et al., 2022a,b; Meade et al., 2022; Deshpande et al., 2023; Feng et al., 2023; Ladhak et al., 2023) or cause errors due by the contamination in the pre-training data as hullucinations (Dziri et al., 2022a,b; McKenna et al., 2023; Ji et al., 2023). This suggests the memorization and inference abilities of PLMs are strongly related, and the pre-trained knowledge of the PLMs influences their inference abilities.