LambdaKG: A Library for Pre-trained Language Model-Based Knowledge Graph Embeddings

Xin Xie[1]*, **Zhoubo Li**[1]*, **Xiaohan Wang**[1]*, **Zekun Xi**[1], **Ningyu Zhang**[1] †
[1] Zhejiang University
{xx2020,zhoubo.li,wangxh07,zhangningyu}@zju.edu.cn,
https://zjunlp.github.io/project/promptkg/

Abstract

Knowledge Graphs (KGs) often have two characteristics: heterogeneous graph structure and text-rich entity/relation information. Text-based KG embeddings can represent entities by encoding descriptions with pre-trained language models, but no open-sourced library is specifically designed for KGs with PLMs at present. In this paper, we present **LambdaKG**, a library for KGE that equips with many pre-trained language models (e.g., BERT, BART, T5, GPT-3) and supports various tasks (e.g., knowledge graph completion, question answering, recommendation, and knowledge probing). **LambdaKG** is publicly open-sourced[1], with a demo video[2] and long-term maintenance.

1 Introduction

Knowledge Graphs (KGs) encode real-world facts as structured data and have drawn significant attention from academia, and industry (Zhang et al., 2022b). Knowledge Graph Embedding (KGE) aims to project the relations and entities into a continuous vector space, which can enhance knowledge reasoning abilities and feasibly be applied to downstream tasks: question answering (Saxena et al., 2022), recommendation (Zhang et al., 2021) and so on (Chen et al., 2022b). Previous **embedding-based** KGE methods, such as TransE (Bordes et al., 2013), involved embedding relational knowledge into a vector space and subsequently optimizing the target object by applying a pre-defined scoring function to those vectors. A few remarkable embedding-based KGE toolkits have been developed, such as OpenKE (Han et al., 2018), LibKGE (Broscheit et al., 2020), PyKEEN (Ali et al., 2021), CogKGE (Jin et al., 2022) and NeuralKG (Zhang et al., 2022c). Nevertheless, these embedding-based KGE approaches are restricted in expressiveness regarding the shallow network architectures without using any side information (e.g., textual description).

By comparison with embedding-based KGE approaches, text-based methods incorporate available texts for KGE. With the development of Pre-trained Language Models (PLMs), many text-based models (Xie et al., 2022; Saxena et al., 2022; Kim et al., 2020; Markowitz et al., 2022; Chen et al., 2022a; Liu et al., 2022) have been proposed, which can obtain promising performance and take advantage of allocating a fixed memory footprint for large-scale real-world KGs. Recently, large language models (LLMs) (e.g., GPT-3 (Brown et al., 2020), ChatGPT (OpenAI, 2022)) further demonstrated the ability to perform a variety of natural language processing (NLP) tasks without adaptation, providing potential opportunities of better knowledge representations. However, there is **no comprehensive open-sourced library particularly designed for KGE with PLMs at present**, which makes it challenging to test new methods and make rigorous comparisons with previous approaches.

In this paper, we share with the community a pre-trained **LA**nguage **M**odel-**B**ase**D** libr**A**ry for **KG**Es and applications called **LambdaKG** (MIT License), which supports various cutting-edge models. Specifically, we equip **LambdaKG** with both small PLMs, e.g., BERT (Devlin et al., 2018; Yao et al., 2019), BART (Lewis et al., 2020; Liu et al., 2021), T5 (Raffel et al., 2020; Saxena et al., 2022); and large PLMs, e.g., GPT-3 (Brown et al., 2020), ChatGPT (OpenAI, 2022), by developing two major paradigms of discrimination-based and generation-based methods for KGEs. **LambdaKG** supports factual and commonsense KGs with diverse tasks, including KG completion, question answering, recommendation, and knowledge probing (LAMA). We will provide maintenance to meet new tasks, new requests and fix bugs.

* Equal contribution and shared co-first authorship.
† Corresponding author.
[1]Code: https://github.com/zjunlp/PromptKG/tree/main/lambdaKG
[2]Video: http://deepke.zjukg.cn/lambdakg.mp4

Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics: System Demonstrations, pages 25–33
November 1–4, 2023.

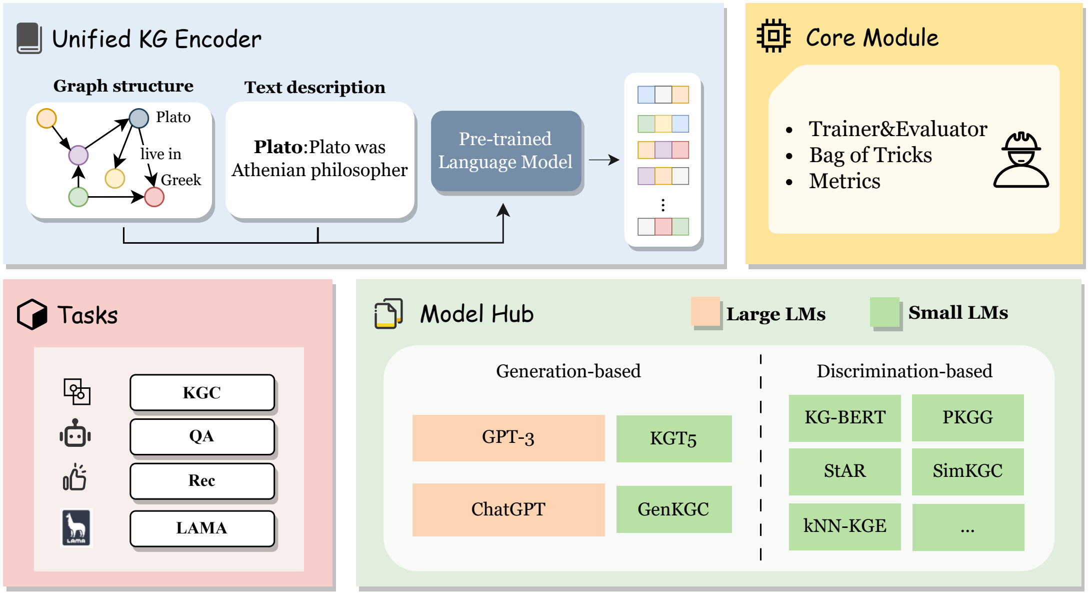

Figure 1: The architecture and features of **LambdaKG**.

2 System Architecture

The overall features & architecture of **LambdaKG** are presented in Figure 1. We will detail two major types of PLM-based KGE methods (discrimination-based and generation-based) with various PLMs.

Our design principles are: 1) *Core Module with Unified KG Encoder*: **LambdaKG** utilizes a unified encoder to pack graph structure and text semantics, with convenient `Trainer&Evaluator`, `Metric`, and `Bag of Tricks`; 2) *Model Hub*: **LambdaKG** is integrated with many cutting-edge PLM-based KGE models; 3) *Flexible Downstream Tasks*: **LambdaKG** disentangles KG representation learning and downstream tasks.

2.1 Core Module

2.1.1 Trainer&Evaluator

Typically, the training process with **LambdaKG** can be decomposed into several distinct steps, such as the forward and backward passes (i.e., `training_step`), logging of intermediate results (`log`), and model evaluation (`evaluate_step`). Our `Trainer` class provides a flexible and modular framework for training different types of models, with customizable functions to handle various tasks, such as computing the loss function and updating model parameters. Moreover, the `Trainer` class allows users to define their own plugins, which can be integrated seamlessly into the training pipeline to provide additional functionalities.

2.1.2 Metric

We design the `Metric` class to evaluate different models for various tasks. Specifically, we use `hits@k` with k values of 1, 3, 10 and *mean rank (MR)* as the evaluation metrics. Hits@k measures the proportion of correct predictions among the top k-ranked results, while MR calculates the average rank of the correct answer. We also implement `BLEU-1` score to evaluate the commonsense KG completion tasks following Hwang et al. (2021).

2.1.3 Bag of Tricks

All models in the **LambdaKG** are based on PLMs, and we equip a bag of tricks of training techniques to improve their performance. In particular, we employ different pluggable modules such as `label smoothing` and `exponential moving average` to assist in the training of models. We implement `early stopping` and `fast run` modules to prevent overfitting with small data by introducing early stopping and automatic verification mechanisms. Furthermore, we integrate an off-the-shelf `Top-k negative sampling strategy` to enhance the training by selecting the most informative negative samples during the training process.

2.2 Unified KG Encoder

Since **LambdaKG** is based on PLMs, the most critical thing is to convert structural triples into plain natural language for PLMs to understand. We introduce a unified KG encoder to represent graph structure and text semantics, supporting different

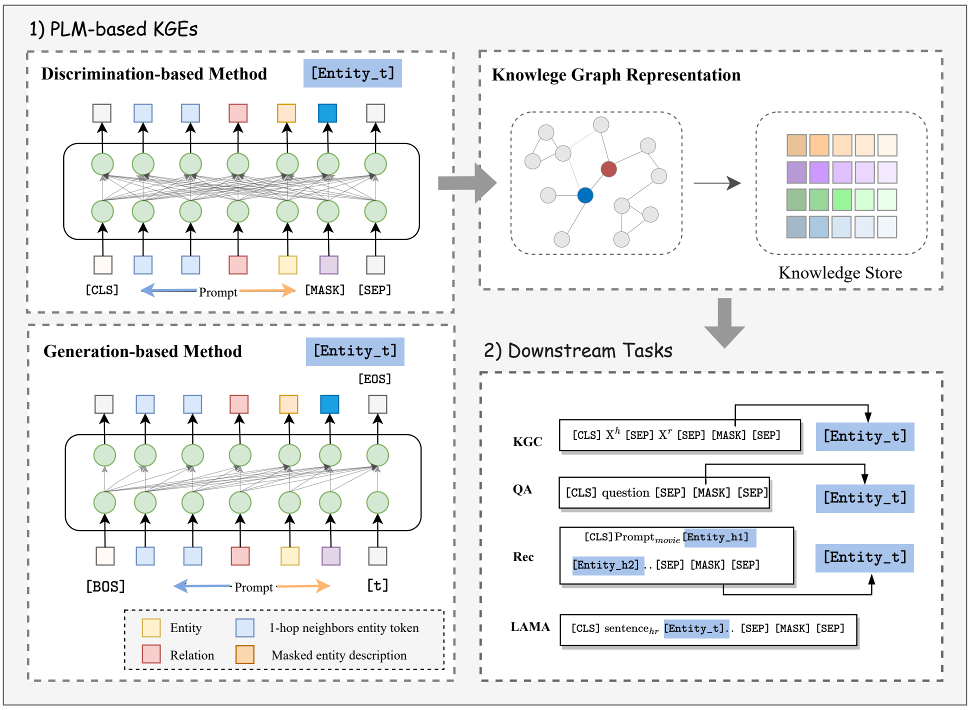

Figure 2: PLM-based KGEs in **LambdaKG** and those KGEs can be applied to KGC, QA, recommendation and knowledge probing. *Entity_t* refers to the target tail entity, answer entity, recommended items, and target tail entity for different tasks, which follows the pre-train (obtain the embedding) and fine-tune paradigm (task-specific tuning).

types of PLM-based KGE methods. To encode the graph structure, we sample 1-hop neighbor entities and concatenate their tokens as input for implicit structure information. With such a unified KG encoder, **LambdaKG** can encode both heterogeneous graph structure and text-rich semantic information. For the discrimination-based method, the input is built on the plain text description:

$$\begin{aligned} X_{\text{hr pair}} &= \texttt{[CLS]}\ X^h \texttt{[SEP]}\ X^r\ \texttt{[SEP]} \\ X_{\text{tail}} &= \texttt{[CLS]}\ X^t\ \texttt{[SEP]}. \end{aligned} \tag{1}$$

where X^h, X^r, and X^t refer to the text sequence of the head entity, relation, and tail entity, respectively. Referring to some prompt learning methods like kNN-KGE (Zhang et al., 2022a), we represent entities and relations in KG with special tokens (See §2.3) and obtain the input as:

$$X = \texttt{[CLS]} X^h \texttt{[Entity h]}\ \texttt{[SEP]}\ X^r\ \texttt{[SEP]}\ \texttt{[MASK]}\ \texttt{[SEP]}, \tag{2}$$

where `[Entity h]` represents the special token to the head entity.

For the generation-based model, we leverage the tokens in X^h and X^r to optimize the model with the label X^t. When predicting the head entity, we add a special token `[reverse]` in the input sequence for reverse reasoning.

2.3 Model Hub

As shown in Figure 2 and Table 1, **LambdaKG** consists of a `Model Hub` which supports many representative PLM-based KGE methods, mainly follow the two major paradigms of discrimination-based methods and generation-based methods as:

Discrimination-based methods There are three kinds of models based on the discrimination method: the first one (e.g., KG-BERT (Yao et al., 2019), PKGC (Lv et al., 2022)) utilizes a single encoder to encode triples of KG with text description; another kind of model (e.g., StAR (Wang et al., 2021), SimKGC (Wang et al., 2022)) leverages siamese encoder (two-tower models) with PLMs to encode entities and relations respectively. For the first kind, the score of each triple is expressed as:

$$\text{Score}(h, r, t) = \text{TransformerEnc}(X^h, X^r, X^t), \tag{3}$$

where TransformerEnc is the BERT model followed by a binary classifier. However, these models have to iterate all the entities calculating scores to decide the correct one, which is computation-intensive, as shown in Table 1. In contrast, two-tower models like StAR (Wang et al., 2021) and SimKGC (Wang et al., 2022) usually encode $\langle h, r\rangle$ and t to obtain the embeddings. Then, they use a

Model	PLM	Support Tasks	Complexity
KGBERT (Yao et al., 2019)	MLM	KGC	$\mathcal{O}(\lvert L\rvert^2\lvert\mathcal{E}\rvert^2\lvert\mathcal{R}\rvert)$
StAR (Wang et al., 2021)	MLM	KGC	$\mathcal{O}(\lvert L/2\rvert^2\lvert\mathcal{E}\rvert(1+\lvert\mathcal{R}\rvert))$
SimKGC (Wang et al., 2022)	MLM	KGC	$\mathcal{O}(\lvert L/2\rvert^2\lvert\mathcal{E}\rvert(1+\lvert\mathcal{R}\rvert))$
kNN-KGE (Zhang et al., 2022a)	MLM	KGC, LAMA	$\mathcal{O}(\lvert L\rvert^2\lvert\mathcal{E}\rvert\lvert\mathcal{R}\rvert)$
KGT5 (Saxena et al., 2022)	Seq2Seq	KGC, QA	$\mathcal{O}(\lvert L/2\rvert^3\lvert\mathcal{E}\rvert\lvert\mathcal{R}\rvert)$
GenKGC (Xie et al., 2022)	Seq2Seq	KGC, QA	$\mathcal{O}(\lvert L/2\rvert^3\lvert\mathcal{E}\rvert\lvert\mathcal{R}\rvert)$

Table 1: Comparison of different methods based on small PLMs. $|L|$ is the length of the triple description. $|L/2|$ can be seen as the length of entity tokens. $|\mathcal{E}|$ and $|\mathcal{R}|$ are the numbers of all unique entities and relations in the graph respectively.

score function to predict the correct tail entity from the candidates, denoted by:

$$\text{Score}(\langle h, r\rangle, t) = \cos(e_{\langle h,r\rangle}, e_t). \tag{4}$$

The final kind of model, e.g., kNN-KGE (Zhang et al., 2022a), utilizes masked language modeling for KGE, which shares the same architecture as normal discrimination PLMs. Note that there are two modules in the normal PLMs: a *word embedding layer* to embed the token ids into semantic space and an *encoder* to generate context-aware token embedding. Here, we take the masked language model and **treat entities and relations as special tokens** in the "word embedding layer". As shown in Figure 2, the model predicts the correct tail entity with the sequence of the head entity and relation token and their descriptions. For the entity/relation embedding, we freeze the *encoder layer*, only tuning the *entity embedding layer*, to optimize the loss function:

$$\mathcal{L} = -\frac{1}{|\mathcal{E}|}\sum_{e_j\in\mathcal{E}} \mathbb{I}(e_j = e_i)\log p\left(\texttt{[MASK]} = e_j \mid X^i; \Theta\right), \tag{5}$$

where Θ represents the parameters of the model, X^i and e_i is the description and the embedding of entity i.

Generation-based methods Generation-based models formulate KG completion or other KG-intensive tasks as sequence-to-sequence generation. Given a triple with the tail entity missing $(h, r, ?)$, models are fed with $\langle X^h, X^r\rangle$ and then output X^t. In the training procedure, generative models maximize the conditional probability:

$$\text{Score}(h, r, t) = \textstyle\prod_{i=1}^{|X^t|} p(x_i^t | x_1^t, x_2^t, ..., x_{i-1}^t; \langle X^h, X^r\rangle). \tag{6}$$

To guarantee the consistency of decoding sequential schemas and tokens in KG, GenKGC (Xie et al., 2022) proposes an entity-aware hierarchical

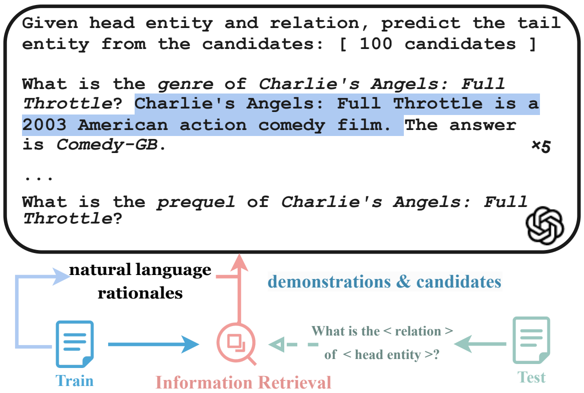

Figure 3: LLM-based KGC. The prompt comprises three components, namely the task description with candidates, demonstrations, and test information.

decoder to constrain X^t. Besides, KGT5 (Saxena et al., 2022) proposes to pre-train generation-based PLMs with text descriptions for KG representation.

LLMs We further apply the LLMs, namely GPT-3 and ChatGPT, to assess their effectiveness in KGE (KGC with link prediction). Generative LLMs allow the KGC task to be framed as input sentences containing header entities and relations, making it easier for the model to generate sentences with tail entities. A well-designed prompt can improve the performance of LLMs, and prior studies indicate incorporating in-context learning can improve accuracy and ensure consistent output. Thus, we adopt a similar approach that the prompt comprises three components: task description with candidates, demonstrations, and test information.

As shown in Figure 3, we employ information retrieval (BM25) to select the top 100 most relevant entities from the training set as candidates. Likewise, the prompt's demonstrations utilize the top-5 most similar instances, which assist the model in comprehending the task more effectively. Furthermore, taking inspiration from the Chain-of-

Thought (CoT) method in reasoning tasks, we utilize `natural language rationales` to improve the model's capacity to reason and explain predictions, ultimately improving its overall performance in KGC tasks. Comparatively, the prompt used for ChatGPT solely utilizes a few demonstrations and test data with these strategies.

2.4 Pluggable KGE for Downstream Tasks

We introduce the technical details of applying KGE to downstream tasks as shown in Figure 2. For knowledge graph completion, we feed the model with the textual information $\langle X^h, X^r \rangle$ of the head entity and the relation, then obtain the target tail entity via mask token prediction. For question answering, we feed the model with the question written in natural language concatenated with a `[MASK]` token to obtain the special token of the target answer (entity). For recommendation, we take the user's interaction history as sequential input (Sun et al., 2019) with entity embeddings and then leverage the mask token prediction to obtain recommended items. For the knowledge probing task, we adopt entity embedding as additional knowledge following PELT (Ye et al., 2022).

3 System Usage

The proposed system can be used in three scenarios. First, users can utilize **LambdaKG** to obtain **PLM-based KGE for knowledge discovery**. `LitModel` serves as the training of link prediction task class and fit for all models in Model Hub. Users can choose proper models in `ModelModule` and specific metrics in `DataModule` to train models to obtain the embedding in the KGs. Moreover, users can utilize **LambdaKG PLM-based KGE for downstream tasks**. We provide various prompts to obtain the knowledge (entity) embedding in KGs for downstream tasks. For different tasks, we design different base classes for users to efficiently implement their own tasks. Finally, we provide an online interactive demo for PLM-based KGE at `https://zjunlp.github.io/project/promptkg/demo.html`.

4 Evaluation

4.1 Knowledge Graph Completion

For the KG completion task with small PLMs, we conduct link prediction experiments on two datasets WN18RR (Dettmers et al., 2018), and FB15k-237 (Toutanova et al., 2015). From Table 2,

Task	Dataset	Method	hits1	MRR
KG Completion	WN18RR	KG-BERT$^\diamond$	4.1	21.6
		StAR$^\diamond$	24.3	40.1
		SimKGC	42.5	60.8
		KGT5	17.9	-
		GenKGC	39.6	-
		kNN-KGE	52.5	57.9
	FB15k-237	KG-BERT$^\diamond$	-	-
		StAR$^\diamond$	20.5	29.6
		SimKGC	22.6	30.1
		KGT5	10.8	-
		GenKGC	19.2	-
		kNN-KGE	28.0	37.3
Question Answering	MetaQA	GT query$^\diamond$	63.3	-
		PullNet$^\diamond$	65.1	-
		KGT5	67.8	-
Recommendation	ML-20m	BERT4Rec$^\diamond$	34.4	47.9
		LambdaKG	37.3	50.5
Knowledge Probing	TREx	BERT	28.6	37.7
		RoBERTa	19.9	27.8
		LambdaKG (RoBERTa)	22.1	29.8
	Squad	BERT	13.2	23.5
		RoBERTa	13.4	24.6
		LambdaKG (RoBERTa)	-	-
	Google RE	BERT	10.3	17.3
		RoBERTa	7.6	12.8
		LambdaKG (RoBERTa)	8.1	14.2

Table 2: Hits1 and MRR (%) results on KGC, question answering, recommendation and knowledge probing tasks. $\diamond$ refers to the results from origin papers.

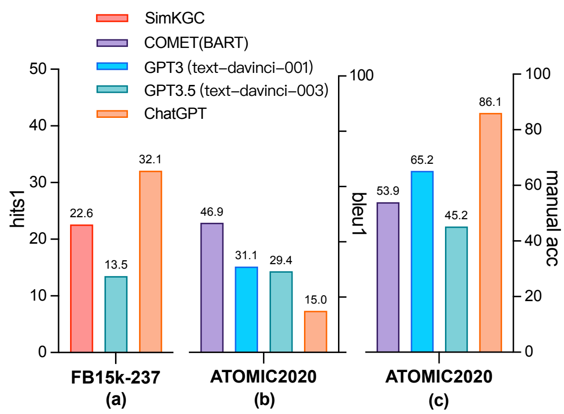

Figure 4: Results on small and large LMs. (a) hit@1 scores on FB15k-237. (b) BLEU-1 scores on ATOMIC2020. (c) Accuracy scores on ATOMIC2020 by manual evaluation.

we observe that the discrimination-based method SimKGC (Wang et al., 2022) (previous state-of-the-art) achieves higher performance than other baselines. Generation-based models like KGT5 (Saxena et al., 2022) and GenKGC (Xie et al., 2022) also yield comparable results and show potential abilities in KG representation.

Small vs. Large LMs We adopt GPT-3/3.5 (`text-davinci-001/003` and ChatGPT) for evaluation and assessment through the interfaces provided by OpenAI. The evaluation of ChatGPT

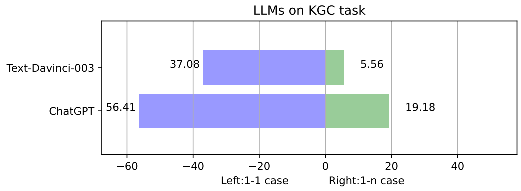

Figure 5: hit@1 of ChatGPT and `text-davinci-003` in FB15k-237.

is conducted on 224 instances, with each relation in the test set. As shown in Figure 4(a), ChatGPT demonstrates better performance, while `text-davinci-003` exhibits a slight gap. The experiment has reaffirmed the capability of LLMs in capturing semantic similarities and regularities among entities, thereby allowing for precise predictions of missing links in knowledge graphs.

In cases where one head entity and relation pair correspond to one or multiple tail entities (1-1 and 1-n cases), we conducted a detailed analysis. Notably, the model performs significantly better in the 1-1 case compared to the 1-n case, as illustrated in Figure 5. Two potential reasons explain this disparity: (1) In the 1-1 case, the model demonstrates a lower propensity for language understanding deviations. Additionally, ChatGPT's training utilizes a larger corpus, enhancing the model to generate accurate responses through analysis and reasoning. (2) The presence of multiple correspondences poses a challenge for the model's capacity to generate informative and contextually relevant responses. Moreover, current evaluation metrics fail to fully capture the intricacy of the responses necessary to properly handle such questions.

We further conduct experiments on **commonsense KG completion** with ATOMIC2020 (Hwang et al., 2021). As suggested in the paper, we sample 5,000 test queries to evaluate the models (excluding ChatGPT). COMET (BART) is fine-tuned through supervised learning and utilizes greedy decoding to generate answers. For GPT3 and ChatGPT, we provide each relation with 5 examples of heads and tails to construct prompts and evaluate them in a zero-shot setting. The results, as shown in Figure 4(b), demonstrates the BLEU-1 scores on the sampled 5,000 queries, while we sample 115 (5 for each relation) queries from the test for ChatGPT. The results indicate that GPT-3 exhibits limited performance in the system evaluation. After analyzing several cases, we sample 115 (5 for each relation) queries as a benchmark and apply manual scoring to evaluate models. Figure 4(c) depicts the accuracy scores of each model. Our study reveals that ChatGPT is capable of generating reasonable outputs, but they are quite different from the ground truth, which accounts for the final results.

4.2 Question Answering

KG is known to be helpful for the task of question answering. We apply **LambdaKG** to question answering and conduct experiments on the MetaQA dataset. Due to computational resource limits, we only evaluate the 1-hop inference performance. From Table 2, KGT5 in **LambdaKG** yields the best performance.

4.3 Recommendation

For the recommendation task, we conduct experiments on a well-established version ML-20m[3]. Linkage of ML-20m and Freebase offered by KB4Rec (Zhao et al., 2019) is utilized to obtain textual descriptions of movies in ML-20m. With movie embeddings pre-trained on these descriptions, we conduct experiments on sequential recommendation tasks following the settings of BERT4Rec (Sun et al., 2019). We notice that **LambdaKG** is confirmed to be effective for the recommendation compared with BERT4Rec.

4.4 Knowledge Probing

Knowledge probing (Petroni et al., 2019) examines the ability of LMs (BERT, RoBERTa, etc.) to recall facts from their parameters. We conduct experiments on LAMA using pre-trained BERT (*bert-base-uncased*) and RoBERTa (*roberta-base*) models. To prove that entity embedding enhanced by KGs helps LMs grab more factual knowledge from PLMs, we train a pluggable entity embedding module following PELT (Ye et al., 2022). As shown in Table 2, the performance boosts while we use the entity embedding module.

5 Conclusion and Future Work

We propose **LambdaKG**, a library that establishes a unified toolkit with well-defined modules and easy-to-use interfaces to support research on using PLMs on KGs. In the future, we will continue to integrate more models and tasks (e.g., dialogue) into the proposed library to facilitate the research progress of the KG.

[3] `https://grouplens.org/datasets/movielens/20m/`

Acknowledgment

We would like to express our heartfelt gratitude to the anonymous reviewers for their thoughtful and kind comments. This work was supported by the National Natural Science Foundation of China (No.62206246), Zhejiang Provincial Natural Science Foundation of China (No. LGG22F030011), Ningbo Natural Science Foundation (2021J190), Yongjiang Talent Introduction Programme (2021A-156-G).

References

Mehdi Ali, Max Berrendorf, Charles Tapley Hoyt, Laurent Vermue, Sahand Sharifzadeh, Volker Tresp, and Jens Lehmann. 2021. Pykeen 1.0: A python library for training and evaluating knowledge graph embeddings. *J. Mach. Learn. Res.*, 22:82:1–82:6.

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

Samuel Broscheit, Daniel Ruffinelli, Adrian Kochsiek, Patrick Betz, and Rainer Gemulla. 2020. Libkge - A knowledge graph embedding library for reproducible research. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 165–174. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher 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 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Chen Chen, Yufei Wang, Bing Li, and Kwok-Yan Lam. 2022a. Knowledge is flat: A seq2seq generative framework for various knowledge graph completion. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 4005–4017. International Committee on Computational Linguistics.

Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022b. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pages 2778–2788. ACM.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

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, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pages 139–144. Association for Computational Linguistics.

Jena D. Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. 2021. (comet-) atomic 2020: On symbolic and neural commonsense knowledge graphs. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 6384–6392. AAAI Press.

Zhuoran Jin, Tianyi Men, Hongbang Yuan, Zhitao He, Dianbo Sui, Chenhao Wang, Zhipeng Xue, Yubo Chen, and Jun Zhao. 2022. Cogkge: A knowledge graph embedding toolkit and benchmark for representing multi-source and heterogeneous knowledge. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, ACL 2022 - System Demonstrations, Dublin, Ireland, May 22-27, 2022*, pages 166–173. Association for Computational Linguistics.

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

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, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Xiao Liu, Shiyu Zhao, Kai Su, Yukuo Cen, Jiezhong Qiu, Mengdi Zhang, Wei Wu, Yuxiao Dong, and Jie Tang. 2022. Mask and reason: Pre-training knowledge graph transformers for complex logical queries. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, pages 1120–1130. ACM.

Ye Liu, Yao Wan, Lifang He, Hao Peng, and Philip S. Yu. 2021. KG-BART: knowledge graph-augmented BART for generative commonsense reasoning. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 6418–6425. AAAI Press.

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, Dublin, Ireland, May 22-27, 2022*, pages 3570–3581. Association for Computational Linguistics.

Elan Markowitz, Keshav Balasubramanian, Mehrnoosh Mirtaheri, Murali Annavaram, Aram Galstyan, and Greg Ver Steeg. 2022. StATIK: Structure and text for inductive knowledge graph completion. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 604–615, Seattle, United States. Association for Computational Linguistics.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue. `https://openai.com/blog/chatgpt/`.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 2463–2473. Association for Computational Linguistics.

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.*, 21:140:1–140:67.

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), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 2814–2828. Association for Computational Linguistics.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, pages 1441–1450. ACM.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1499–1509. The Association for Computational Linguistics.

Bo Wang, Tao Shen, Guodong Long, Tianyi Zhou, Ying Wang, and Yi Chang. 2021. Structure-augmented text representation learning for efficient knowledge graph completion. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pages 1737–1748. ACM / IW3C2.

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), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 4281–4294. 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. In *Companion of The Web Conference 2022, Virtual Event / Lyon, France, April 25 - 29, 2022*, pages 162–165. ACM.

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

Deming Ye, Yankai Lin, Peng Li, Maosong Sun, and Zhiyuan Liu. 2022. A simple but effective pluggable entity lookup table for pre-trained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 523–529. Association for Computational Linguistics.

Ningyu Zhang, Qianghuai Jia, Shumin Deng, Xiang Chen, Hongbin Ye, Hui Chen, Huaixiao Tou, Gang

Huang, Zhao Wang, Nengwei Hua, and Huajun Chen. 2021. Alicg: Fine-grained and evolvable conceptual graph construction for semantic search at alibaba. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, pages 3895–3905. ACM.

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

Ningyu Zhang, Xin Xu, Liankuan Tao, Haiyang Yu, Hongbin Ye, Shuofei Qiao, Xin Xie, Xiang Chen, Zhoubo Li, Lei Li, et al. 2022b. Deepke: A deep learning based knowledge extraction toolkit for knowledge base population. In *Proceedings of EMNLP Demonstration*.

Wen Zhang, Xiangnan Chen, Zhen Yao, Mingyang Chen, Yushan Zhu, Hongtao Yu, Yufeng Huang, Yajing Xu, Ningyu Zhang, Zezhong Xu, Zonggang Yuan, Feiyu Xiong, and Huajun Chen. 2022c. Neuralkg: An open source library for diverse representation learning of knowledge graphs. In *SIGIR*, pages 3323–3328. ACM.

Wayne Xin Zhao, Gaole He, Kunlin Yang, Hongjian Dou, Jin Huang, Siqi Ouyang, and Ji-Rong Wen. 2019. Kb4rec: A data set for linking knowledge bases with recommender systems. *Data Intell.*, 1(2):121–136.