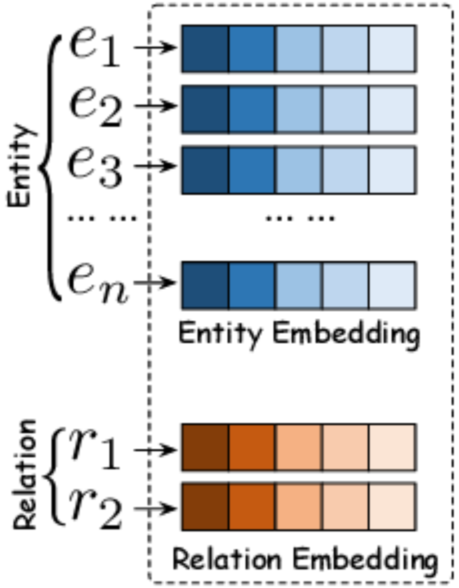
LambdaKG: A Library for Pre-trained Language Model-Based Knowledge Graph Embeddings

Xin Xie*, Zhoubo Li*, Xiaohan Wang*, Yuqi Zhu*, Ningyu Zhang, Jintian Zhang,
Siyuan Cheng Bozhong Tian, Shumin Deng, Feiyu Xiong, Huajun Chen

Pretrained Language Model-based KGEs

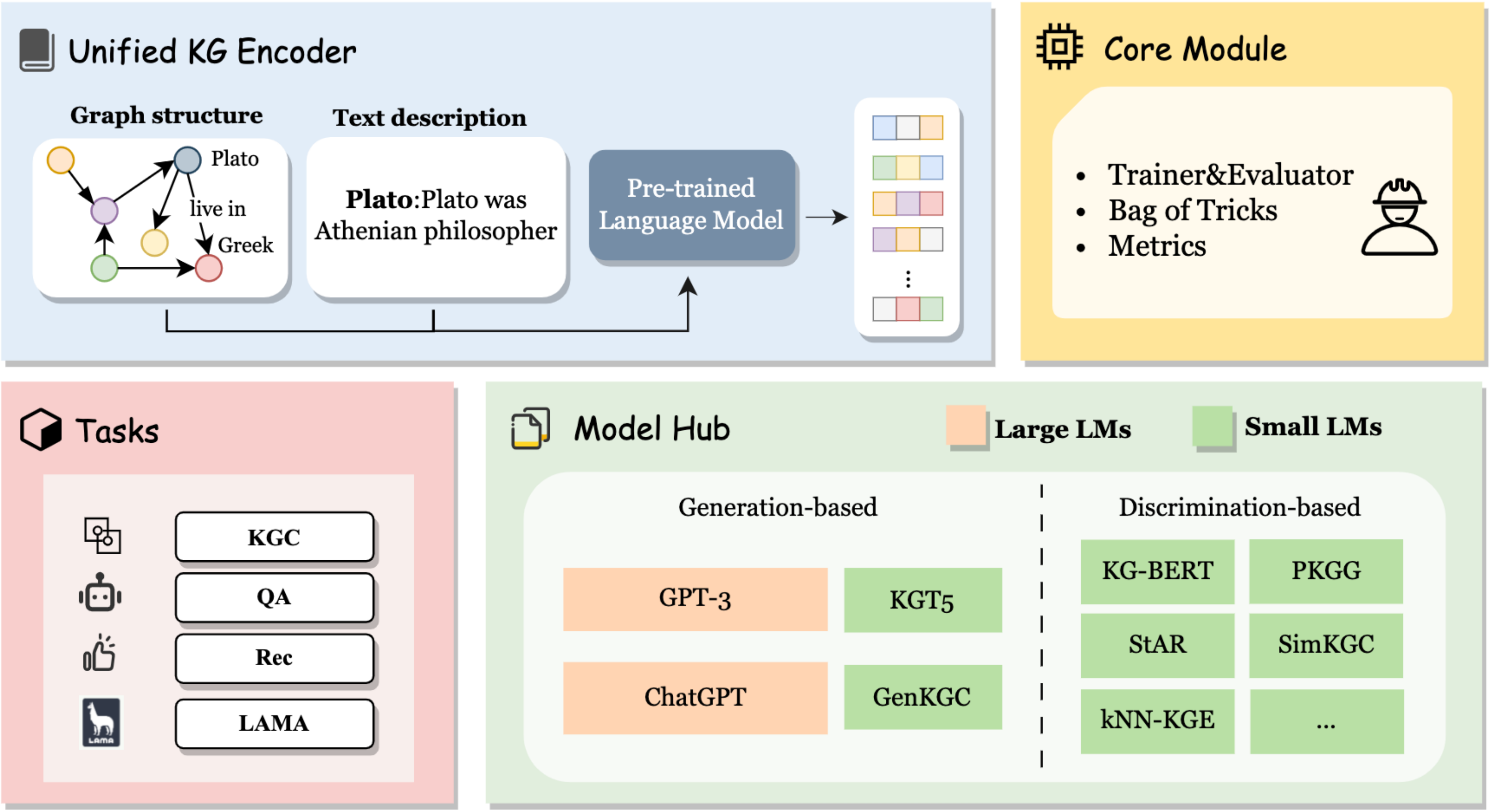
Knowledge Graph

Pretrained Language Models
BERT、T5、GPT-3…..



Knowledge Graph Embedding

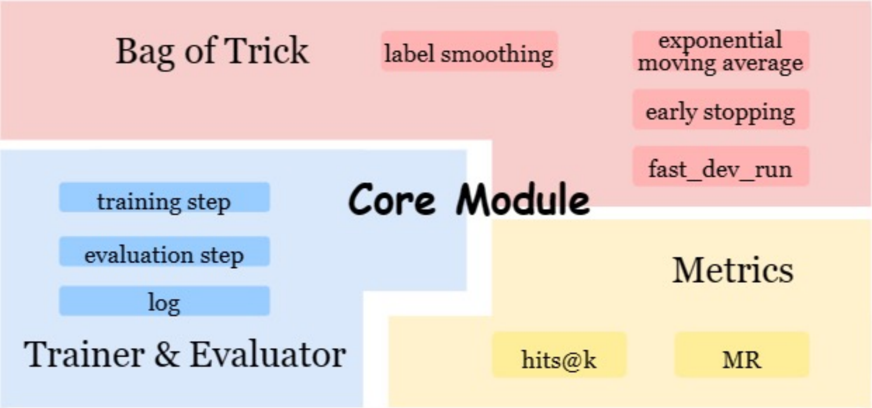
LambdaKG: Features & Architecture

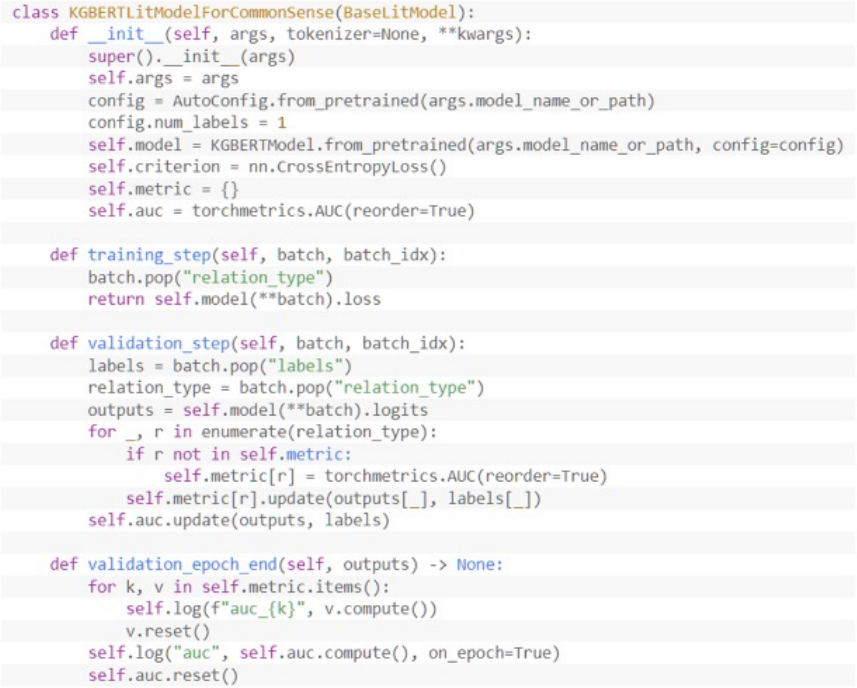


- Bag of Tricks
- Large and Small PLMS
- Flexible Downstream Task

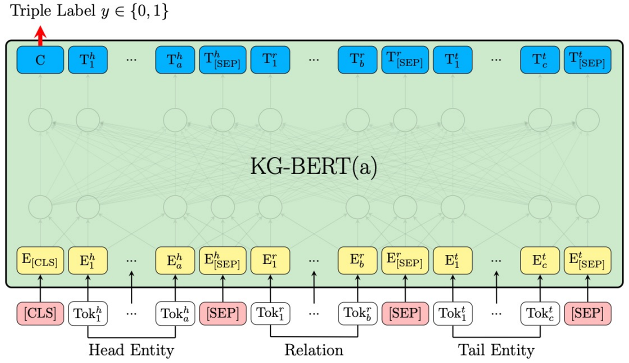
Discrimination-based and Generation-based Methods

浙江大学 ZHEJIANG UNIVERSITY

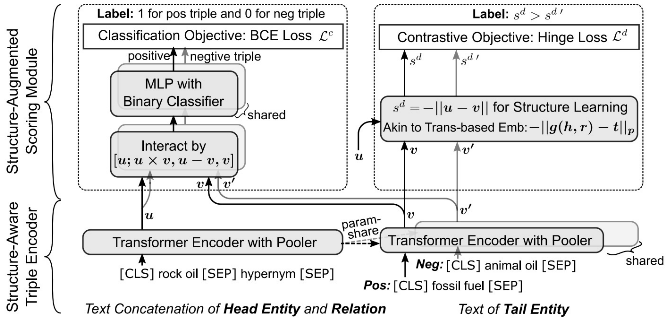




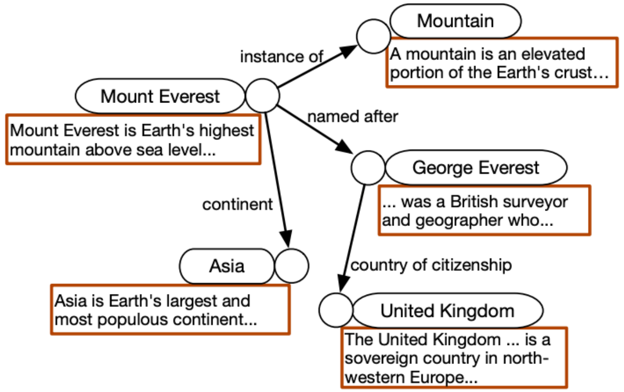
```
class KGBERTLitModelForCommonSense(BaseLitModel):
    def __init__(self, args, tokenizer=None, **kwargs):
        super().__init__(args)
        self.args = args
        config = AutoConfig.from_pretrained(args.model_name_or_path)
        config.num_labels = 1
        self.model = KGBERTModel.from_pretrained(args.model_name_or_path, config=config)
        self.criterion = nn.CrossEntropyLoss()
        self.metric = {}
        self.auc = torchmetrics.AUC(reorder=True)

    def training_step(self, batch, batch_idx):
        batch.pop("relation_type")
        return self.model(**batch).loss

    def validation_step(self, batch, batch_idx):
        labels = batch.pop("labels")
        relation_type = batch.pop("relation_type")
        outputs = self.model(**batch).logits
        for _, r in enumerate(relation_type):
            if r not in self.metric:
                self.metric[r] = torchmetrics.AUC(reorder=True)
            self.metric[r].update(outputs[_], labels[_])
        self.auc.update(outputs, labels)

    def validation_epoch_end(self, outputs) -> None:
        for k, v in self.metric.items():
            self.log(f"auc_{k}", v.compute())
            v.reset()
        self.log("auc", self.auc.compute(), on_epoch=True)
        self.auc.reset()
```



KG-BERT Arxiv' 20


Structure-Augmented Scoring Module
Structure-Aware Triple Encoder
Label: 1 for pos triple and 0 for neg triple
Classification Objective: BCE Loss $\mathcal{L}^c$
positive
negtive triple
MLP with Binary Classifier
shared
Interact by $[u; u \times v, u - v, v]$
Label: $s^d > s^{d\prime}$
Contrastive Objective: Hinge Loss $\mathcal{L}^d$
$s^d = -\|u - v\|$ for Structure Learning
Akin to Trans-based Emb: $-\|g(h,r) - t\|_p$
Transformer Encoder with Pooler
param-share
Transformer Encoder with Pooler
shared
[CLS] rock oil [SEP] hypernym [SEP]
Neg: [CLS] animal oil [SEP]
Pos: [CLS] fossil fuel [SEP]
Text Concatenation of Head Entity and Relation
Text of Tail Entity

StAR WWW' 21


Mountain
A mountain is an elevated portion of the Earth's crust...
instance of
Mount Everest
Mount Everest is Earth's highest mountain above sea level...
named after
George Everest
... was a British surveyor and geographer who...
continent
Asia
Asia is Earth's largest and most populous continent...
country of citizenship
United Kingdom
The United Kingdom ... is a sovereign country in north-western Europe...

SimKGC ACL'22

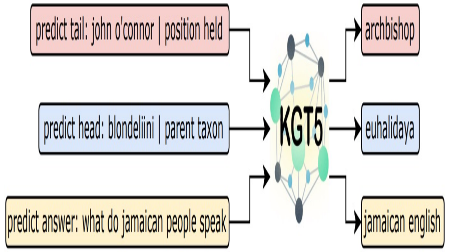
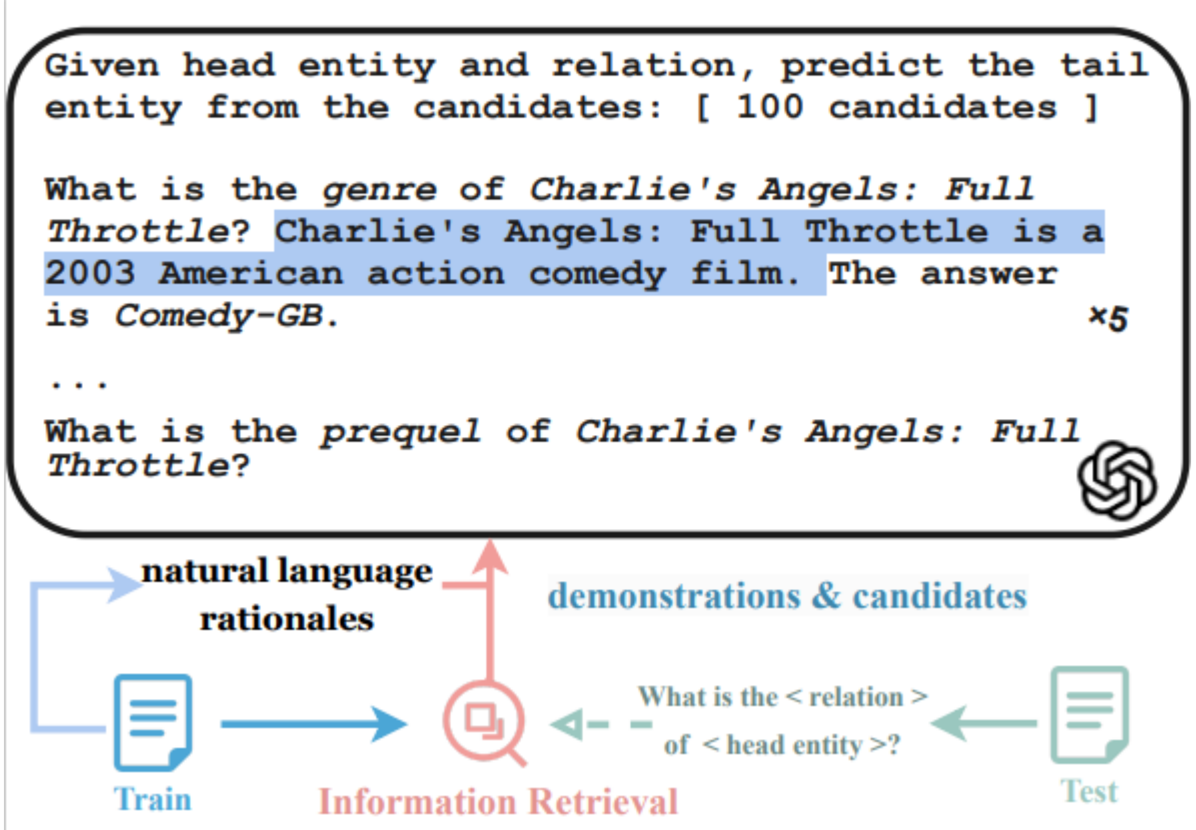

predict tail: john o'connor | position held
archbishop
predict head: blondellini | parent taxon
euhalidaya
predict answer: what do jamaican people speak
jamaican english
KGT5

Figure 1: Overview of our method KGT5. KGT5 is first trained on the link prediction task (predicting head/tail entities, given tail/head and relation). For question answering, the same model is further finetuned using QA pairs.

KGT5 ACL' 22

求是创新

LambdaKG with LLM



Given head entity and relation, predict the tail entity from the candidates: [100 candidates]

What is the *genre* of *Charlie's Angels: Full Throttle*? Charlie's Angels: Full Throttle is a 2003 American action comedy film. The answer is *Comedy-GB*. ×5

...

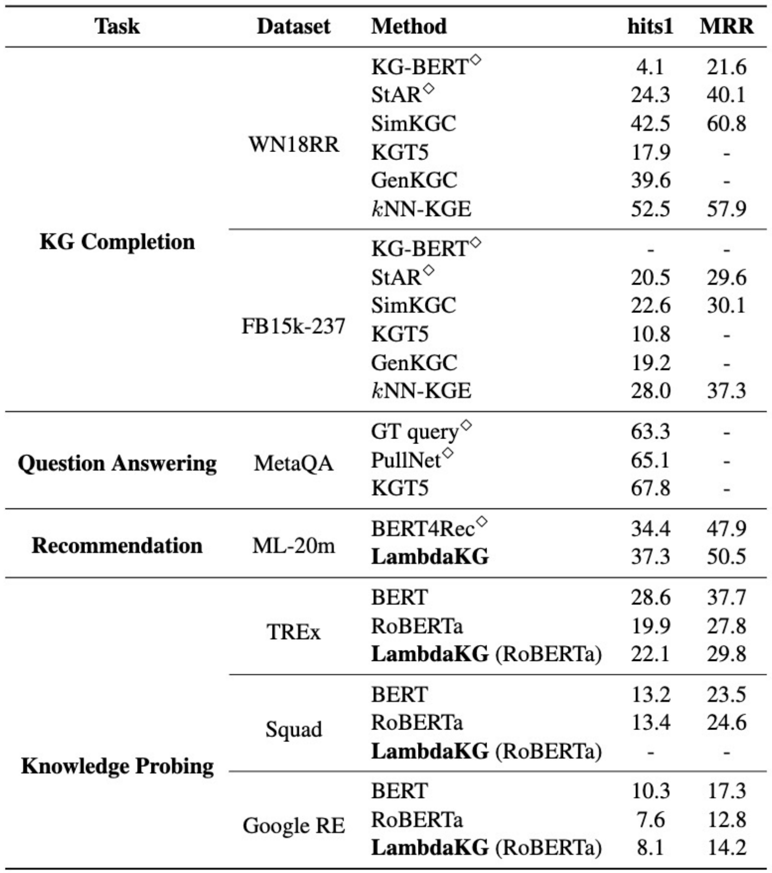
What is the *prequel* of *Charlie's Angels: Full Throttle*?

natural language rationales
demonstrations & candidates
Train
Information Retrieval
What is the < relation > of < head entity >?
Test

LambdaKG provides LLMs, namely **GPT-3** and **ChatGPT**, to evaluate their experimental effectiveness in KGE.

The prompt comprises three components:
task description with **candidates, demonstrations,** and **test information**.

Small vs. Large PLMs and Various KG-based Tasks



Task	Dataset	Method	hits1	MRR
KG Completion	WN18RR	KG-BERT◇	4.1	21.6
		StAR◇	24.3	40.1
		SimKGC	42.5	60.8
		KGT5	17.9	-
		GenKGC	39.6	-
		*k*NN-KGE	52.5	57.9
	FB15k-237	KG-BERT◇	-	-
		StAR◇	20.5	29.6
		SimKGC	22.6	30.1
		KGT5	10.8	-
		GenKGC	19.2	-
		*k*NN-KGE	28.0	37.3
Question Answering	MetaQA	GT query◇	63.3	-
		PullNet◇	65.1	-
		KGT5	67.8	-
Recommendation	ML-20m	BERT4Rec◇	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. ◇ refers to the results from origin papers.

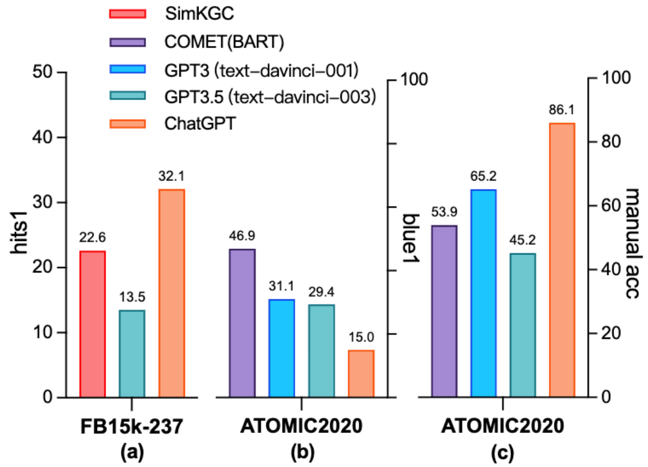


Figure 4: Results on small and large LMs. (a) hit@1 scores on FB15k-237. (b) blue-1 scores on ATOMIC2020. (c) acc scores on ATOMIC2020 by manual evaluation.

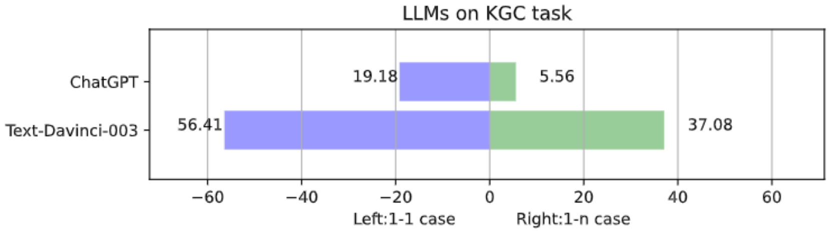
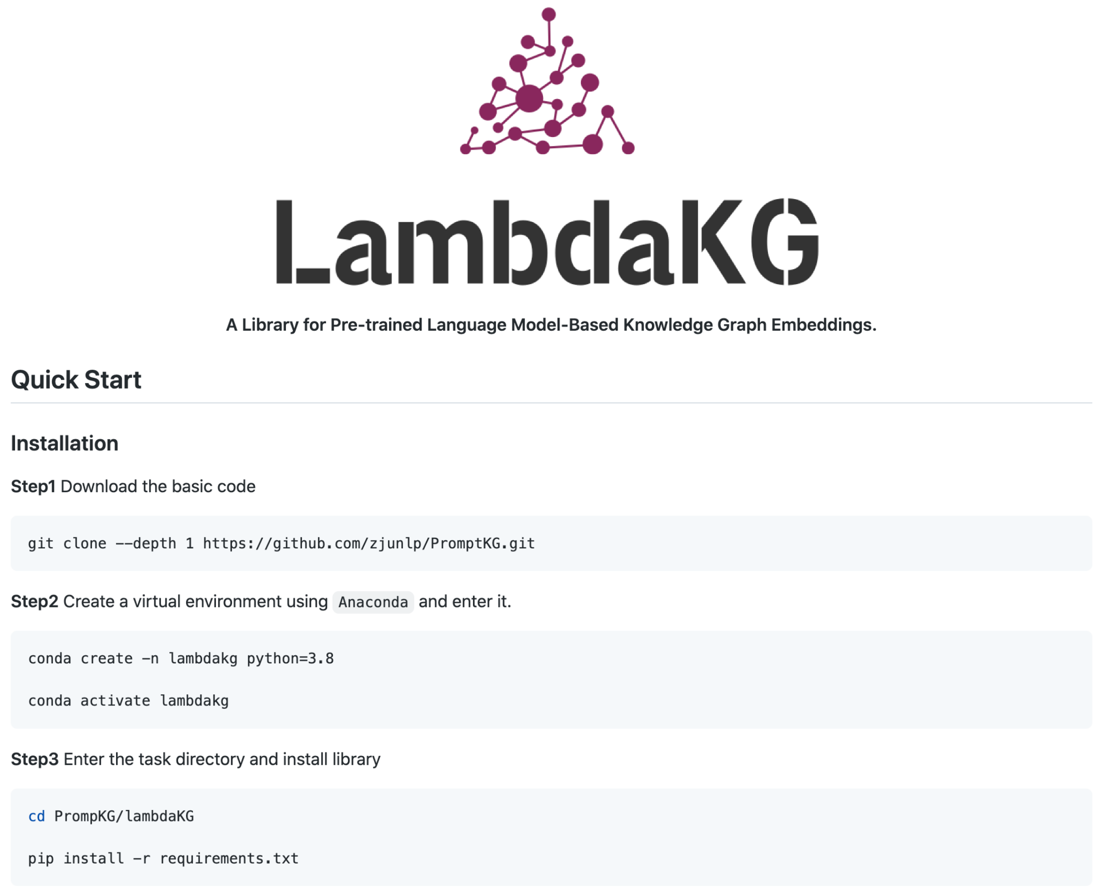


Figure 5: hit@1 of ChatGPT and `text-davinci-003` in FB15k-237.

Open-sourced with an Online System



LambdaKG

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

Quick Start

Installation

Step1 Download the basic code

```
git clone --depth 1 https://github.com/zjunlp/PromptKG.git
```

Step2 Create a virtual environment using `Anaconda` and enter it.

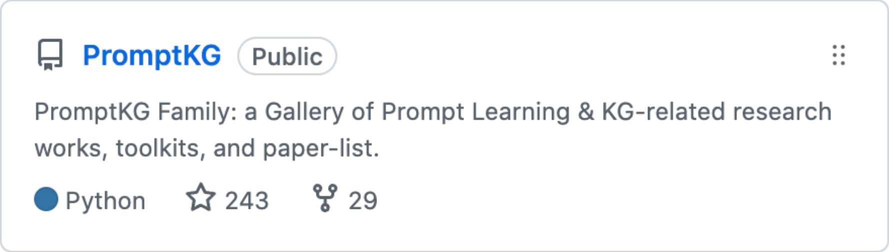
```
conda create -n lambdakg python=3.8

conda activate lambdakg
```

Step3 Enter the task directory and install library

```
cd PrompKG/lambdaKG

pip install -r requirements.txt
```

PyTorch

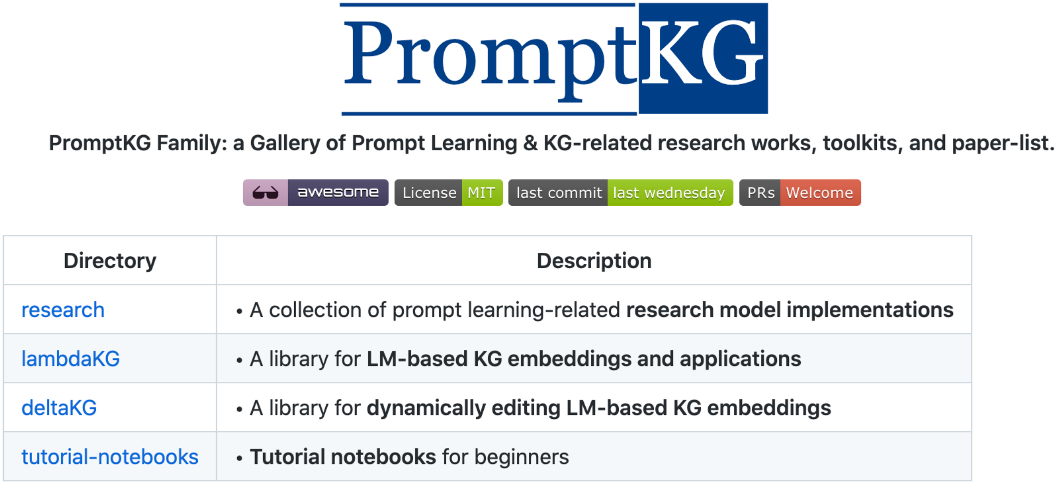
Transformers



PromptKG Public

PromptKG Family: a Gallery of Prompt Learning & KG-related research works, toolkits, and paper-list.

● Python ☆ 243 29



PromptKG

PromptKG Family: a Gallery of Prompt Learning & KG-related research works, toolkits, and paper-list.

Directory	Description
research	• A collection of prompt learning-related **research model implementations**
lambdaKG	• A library for **LM-based KG embeddings and applications**
deltaKG	• A library for **dynamically editing LM-based KG embeddings**
tutorial-notebooks	• **Tutorial notebooks** for beginners

https://zjunlp.github.io/project/promptkg/

浙江大學
ZHEJIANG UNIVERSITY

Thank You!

ACCEPT MY ENDLESS GRATITUDE