KC-GenRe: A Knowledge-constrained Generative Re-ranking Method Based on Large Language Models for Knowledge Graph Completion

Yilin Wang[1], **Minghao Hu**[2,*], **Zhen Huang**[3,*], **Dongsheng Li**[3], **Dong Yang**[3], **Xicheng Lu**[3]

[1] Defense Innovation Institute, Academy of Military Sciences,
[2] Information Research Center of Military Science,
[3] National Key Laboratory of Parallel and Distributed Computing
{wangyilin14, huangzhen, dsli, yangdong14, xclu}@nudt.edu.cn, huminghao16@gmail.com

Abstract

The goal of knowledge graph completion (KGC) is to predict missing facts among entities. Previous methods for KGC re-ranking are mostly built on non-generative language models to obtain the probability of each candidate. Recently, generative large language models (LLMs) have shown outstanding performance on several tasks such as information extraction and dialog systems. Leveraging them for KGC re-ranking is beneficial for leveraging the extensive pre-trained knowledge and powerful generative capabilities. However, it may encounter new problems when accomplishing the task, namely mismatch, misordering and omission. To this end, we introduce **KC-GenRe**, a knowledge-constrained generative re-ranking method based on LLMs for KGC. To overcome the mismatch issue, we formulate the KGC re-ranking task as a candidate identifier sorting generation problem implemented by generative LLMs. To tackle the misordering issue, we develop a knowledge-guided interactive training method that enhances the identification and ranking of candidates. To address the omission issue, we design a knowledge-augmented constrained inference method that enables contextual prompting and controlled generation, so as to obtain valid rankings. Experimental results show that KG-GenRe achieves state-of-the-art performance on four datasets, with gains of up to 6.7% and 7.7% in the MRR and Hits@1 metric compared to previous methods, and 9.0% and 11.1% compared to that without re-ranking. Extensive analysis demonstrates the effectiveness of components in KG-GenRe.

Keywords: Knowledge Graph Completion, Large Language Model, Re-ranking

1. Introduction

Knowledge graph (KG) stores facts in the form of triples, where each of them is represented as (head entity, relation, tail entity), i.e., (e_h, r, e_t). However, KGs are generally incomplete as a large number of facts are missing (West et al., 2014), which hinders the performance of a wide range of applications such as question answering (Saxena et al., 2020; Sun et al., 2022), and recommendation systems (Yang et al., 2022). Knowledge graph completion (KGC) is therefore a critical task to predict missing facts for improving KG completeness.

Recently, large language models (LLMs), e.g. GPT3 (Brown et al., 2020) and LLaMA (Touvron et al., 2023a), have shown excellent performance on various tasks such as information extraction and dialog system (Wang et al., 2023a; Touvron et al., 2023a). Meanwhile, several approaches explore the ability of LLMs to perform KGC (Zhu et al., 2023a; Yao et al., 2023). However, due to the uncontrollable and diverse nature of the generation process, reasoning about missing entities directly for query $(e_h, r, ?)$ from the unfine-tuned LLMs requires manual assistance to match the output with entities in KG (Zhu et al., 2023a). This makes it difficult to automatically obtain answers and perform a comprehensive evaluation on the whole dataset. Although there are methods to train LLMs for KGC (Yao et al., 2023), they do not deviate from previous ones (Chen et al., 2022; Xie et al., 2022b) that are based on small generative language models (LMs), e.g., T5 (Raffel et al., 2020). Furthermore, prior methods for generative KGC in the domain of commonsense knowledge (Bosselut et al., 2019; Yang et al., 2023b) allow to generate new entities not belonging to the given KG, which differs from the conventional KGC task focused here.

Lately, re-ranking using LLMs has received attention on tasks such as information retrieval (Zhu et al., 2023b; Pradeep et al., 2023; Ma et al., 2023). Unlike these approaches for re-ranking documents related to a query, we utilize LLMs for re-ranking given candidate entities to implement KGC. Nevertheless, the task is non-trivial that query is relatively brief, consisting solely of a head entity and a relation. Moreover, there is a paucity of knowledge regarding the candidates, which may be limited to their name labels, thereby increasing the difficulty of the re-ranking process.

Additionally, to the best of our knowledge, existing re-ranking methods for KGC based on LMs all

* Corresponding author

LREC-COLING 2024, pages 9668–9680

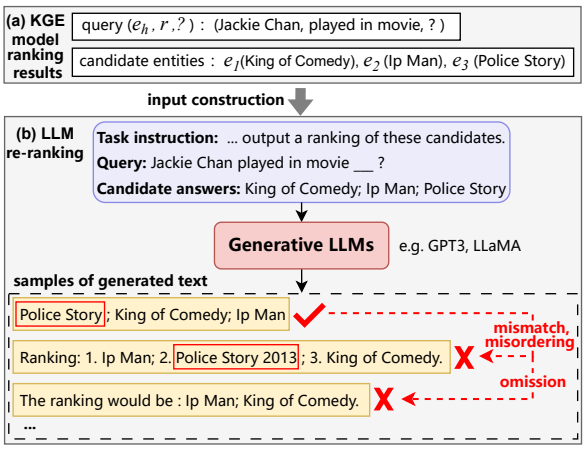


Figure 1: Challenges for KGC re-ranking based on generative LLMs, given query (Jackie Chan, played in movie, ?) and the top-3 candidates, where e_3(Police Story) is the target entity.

utilize non-generative models such as BERT (Devlin et al., 2019) to obtain the probability of each candidate. Although utilizing generative LLMs for re-ranking can inherit their robust capabilities, such as extensive pre-trained knowledge and flexible task reconstruction, there are new challenges that need to be addressed, as exemplified at the bottom of Figure 1: (1) **mismatch**: the generated text contains the candidate entity in KG, but in different textual forms, e.g., generating "Police Story 2013" instead of target "Police Story"; (2) **misordering**: the correct answer is not predicted in the first position; (3) **omission**: the output text fails to include all candidates, particularly the target one. For instance, LLMs may refuse to answer questions by outputting "I'm sorry, but I don't have enough information to answer" (Yao et al., 2023), hindering the performance and evaluation of KGC re-ranking.

To tackle the above problems, we propose **KC-GenRe**, a **k**nowledge-**c**onstrained **gen**erative **re**-ranking method that fully exploits the potential of generative LLMs to perform KGC re-ranking. To overcome the mismatch issue, we formalize the re-ranking task as outputting the order of option identifiers corresponding to these candidates, rather than their names. This eliminates the need for exact text matching of entities and limits the output vocabulary to a known and finite range. To resolve the misordering issue, we design a knowledge-guided interaction training method that utilizes the inference results from the first-stage knowledge graph embedding (KGE) model to enhance the discernment of candidates and learn their relative order. To deal with the omission issue, we present a knowledge-augmented constrained inference method that retrieves contextual knowledge to perform generation under control, so as to output legitimate results. In summary, our contributions are:

- We propose KC-GenRe, a novel knowledge-constrained generative re-ranking model, which is the first to utilize generative LLM for KGC re-ranking as far as we know.
- We design knowledge-guided interactive training and knowledge-augmented constrained inference methods to stimulate the potential of generative LLMs and generate valid ranking of candidates for KGC.
- Experiments on four datasets show that KG-GenRe outperforms start-of-arts results, and extensive analysis demonstrates the effectiveness of the proposed components. Datasets and codes have been open sourced at https://github.com/wylResearch/KC-GenRe.

2. Related Work

2.1. Embedding-based KGC

KGE methods measure the plausibility of triples by learning low-dimensional embeddings of entities and relations. They are popular in implementing KGC task, which can be broadly divided into three categories: distance-based models (Bordes et al., 2013; Qian et al., 2018), semantic matching-based models (Trouillon et al., 2016; Balazevic et al., 2019), and neural network-based models (Schlichtkrull et al., 2018; Dettmers et al., 2018). Auxiliary information such as entity descriptions, hierarchical types are often employed to enhance the embedding so that it contains not only structural but also semantic information (Xie et al., 2016a,b).

2.2. LM-based KGC

LM-based KGC methods typically utilize the textual form of triples, falling into two categories. The first type utilizes LMs to perform KGC independently, which can be transformed into a binary classification task for query triple (e_h, r, e_t)? (Yao et al., 2019; Kim et al., 2020), a matching task to find missing entities for $(e_h, r, ?)$ (Wang et al., 2022), or a text generation task to output target entities (Saxena et al., 2022; Chen et al., 2022; Xie et al., 2022b; Yao et al., 2023; Zhu et al., 2023a). KG-S2S (Chen et al., 2022) trains a small LM, specifically T5 (Raffel et al., 2020), to generate a single candidate entity during each inference process for $(e_h, r, ?)$. Although KG-LLM (Yao et al., 2023) fine-tunes LLMs like LLaMA (Touvron et al., 2023a) to further achieve triple classification and relation prediction tasks, it solely designs input-output templates. Zhu et al. (2023a) employ LLMs for zero-shot and few-shot KGC without fine-tuning, but they need manual sampling for answer acquisition and evaluation.

Above methods linearize the triple knowledge into text, resulting in a lack of structural information.

Hence, the second category combines LMs and KGE models to capture both semantic and structural knowledge. They either integrate the LM into KGE methods (Zhang et al., 2020; Wang et al., 2021) or utilize it as an independent re-ranking stage following KGE model (Lovelace et al., 2021; Lv et al., 2022).

2.3. LM-based KGC Re-ranking

Existing re-ranking methods are basically built on non-generative LMs to compute the probability of each candidate. CEAR (Kolluru et al., 2021) applies BERT (Devlin et al., 2019) to score a set of candidates together by concatenating their textual forms with query $(e_h, r, ?)$. (Lovelace et al., 2021) develops a BERT-based student network guided by a first-stage KGE model to score the text consisting of the query and a candidate. In addition to the query triple (e_h, r, e_t), PKGC (Lv et al., 2022) takes entity definition and attribute as prompts, whose relation templates are manually designed. TAGREAL (Jiang et al., 2023) proposes to automatically generate query prompts and retrieve related information from large corpora to construct input. Different from them, to the best of our knowledge, we are the first to model the KGC re-ranking process utilizing generative LMs, and our contextual knowledge are retrieved from training set.

2.4. Re-ranking Tasks with LLMs

Recently, generative LLMs show generalized and superior performance across numerous tasks such as translation, dialogue, and information extraction (Wang et al., 2023a; Touvron et al., 2023a). It exhibits various capabilities such as zero-shot reasoning, in-context learning, and instruction understanding (Kojima et al., 2022; Min et al., 2022; Ouyang et al., 2022). Meanwhile, in addition to traditional full parameter fine-tuning approach, many techniques have emerged, such as adaptation, prompt learning, and instruction tuning (Hu et al., 2022; Liu et al., 2022; Wang et al., 2023a), so as to apply LLMs to various downstream tasks. Although there are several methods based on LLMs for re-ranking on information retrieval task (Zhu et al., 2023b; Pradeep et al., 2023; Ma et al., 2023), as far as we know that we are the first to fine-tune LLMs for re-ranking on KGC task. With respect to the challenges faced by KGC, we further propose novel training and inference methods.

3. KGC Re-ranking Formulation

Let $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{F})$ be a KG, where $\mathcal{E}$ is the set of entities, $\mathcal{R}$ is the set of relations, and $\mathcal{F} = \{(e_h, r, e_t)|e_h \in \mathcal{E}, r \in \mathcal{R}, e_t \in \mathcal{E}\}$ is the set of facts. Here, e_h and e_t are the head entity and tail entity in a factual triple. Each entity $e \in \mathcal{E}$ and each relation $r \in \mathcal{R}$ has its own labeled name text, which is a sequence of tokens, denoted as $x^e = (w_1^e, w_2^e, \ldots, w_{|x_e|}^e)$ and $x^r = (w_1^r, w_2^r, \ldots, w_{|x_r|}^r)$, where $|x_e|$ and $|x_r|$ are the numbers of tokens in x^e and x^r. Given a query $(e_h, r, ?)$, link prediction task ranks each entity by calculating its score that makes the query hold, so as to achieve KGC.

In the two-stage framework, the first ranking stage typically employs efficient KGE methods to obtain scores for each entity in answering the query $(e_h, r, ?)$. We denote the top-K predicted candidate entities as $E_c = \{e_{t_1}, e_{t_2}, \ldots, e_{t_K}\}$, and their corresponding scores as $S_c = \{s_1, s_2, \ldots, s_K\}$. At the second stage of re-ranking, these promising entities in E_c are converted into a sequence along with the query, which serves as the input to KC-GenRe to output their ranking.

4. KC-GenRe: The Proposed Method

To harness the capabilities of generative LLMs for KGC, we propose a knowledge-constrained generative re-ranking model, named KC-GenRe, to obtain the ranking of top-K predicted candidate entities for a given query $(e_h, r, ?)$, as shown in Figure 2. In Sec. 4.1, we formally introduce the generative KGC re-ranking task proposed here. To achieve comprehensive discernment of candidates and learn their relative ranking, we develop the knowledge-guided interactive training method in Sec. 4.2. For generating effective and legitimate candidate ranking, we design the knowledge-augmented constrained inference method to provide supporting contextual prompt and generation control in Sec. 4.3.

4.1. Generative Re-ranking

Input Composition Given a query $(e_h, r, ?)$ and top-K candidate entities E_c predicted by the first-stage KGE models, we first convert them into input sequence x_{in} by an instruction template T_{in}, i.e., $x_{in} = T_{in}(x_q, x_c)$, where x_q and x_c are query sequence and candidate sequence respectively. We can obtain x_q either by directly concatenating the text of e_h and r, or by adopting pre-constructed prompt, e.g., x_q could be "Jackie Chan played in movie ___?" for query (Jackie Chan, played in movie, ?). As for candidate sequence x_c, it is composed of the given candidate entities E_c with each one equipped with an option identifier selected from the set of option identifiers O, such as "A. Ip Man B. King of Comedy C. Police Story".

Generation Target To address the problem of mismatch, our objective is to generate the order of option identifiers associated with these candidates, instead of their labeled name texts. It restricts the

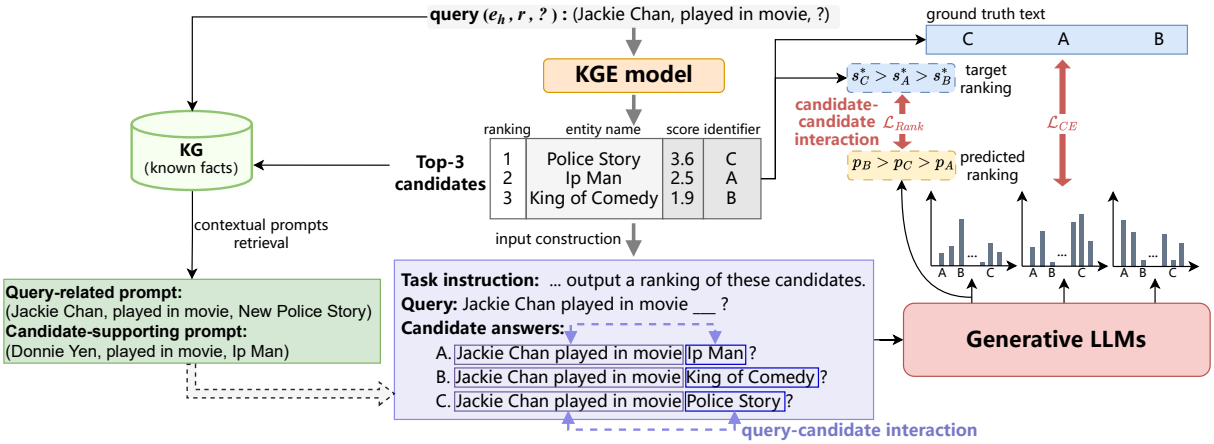


Figure 2: Overview of KC-GenRe, which re-ranks Top-3 candidates predicted by the first-stage KGE model through LLMs for a given query $(e_h, r, ?)$. Its knowledge-guided interactive training method includes query-candidate interaction and candidate-candidate interaction modules, while its knowledge-augmented constrained inference method includes query-related prompt, candidate-supporting prompt, and constrained option generation modules(omitted in the figure).

output vocabulary to a small and fixed set of candidate identifiers, eliminating text matching with a large number of entities. The target output text y is a concatenation of identifiers, e.g., "C A B".

4.2. Knowledge-guided Interactive Training

To address the issue of misordering, we design a knowledge-guided interactive training method, i.e., query-candidate interaction and candidate-candidate interaction, to achieve a thorough identification of candidates and accurately learn their relative ranking by leveraging knowledge from the first-stage KGE models.

Query-Candidate Interaction To enhance the discernment of candidates as answers to the query, we explicitly integrate each candidate with the query to form candidate triple in candidate sequence x_c, rather than just listing candidate entities. Formally, we obtain candidate triple sequence $x_{hrt_i} (i \in \{1, 2, \ldots, K\})$ for candidate entity $e_{t_i} \in E_c$ by populating its name label to the places where entity is missing in the query sequence x_q, e.g., x_{hrt_1} = "Jackie Chan played in movie Ip Man?". Hence, the candidate sequence x_c in Sec. 4.1 would be modified to "A. Jackie Chan played in movie Ip Man? B. Jackie Chan played in movie King of Comedy? C. Jackie Chan played in movie Police Story?".

With explicit query-candidate interaction, the model can directly learn the rationality of each candidate triple that represents as a sequence piece in x_c. In addition, it can establish shorter dependency relationships between the query and each candidate, especially when K is large.

Candidate-Candidate Interaction Different from question answering task that only requires identifying correct answers from given candidates, the goal of this paper is to output the sorted result of all candidates. To learn their relative ranking, we propose to augment their interaction by utilizing the knowledge from first stage, i.e., candidate scores S_c, for learning a ranking loss between the target and the predicted sorts. First, we employ min-max scaling to normalize $S_c = \{s_1, \ldots, s_K\}$ to the range [0,1] as labeled candidate probabilities $S^* = \{s_1^*, \ldots, s_K^*\}$, where s_i^* is the target probability of e_{t_i}. Next, to get the probability predicted by LLMs like LLaMA (Touvron et al., 2023a), we identify the position of the first option identifier in the generated sequence, from which the model's logits for all option identifiers are taken out as the logits of corresponding candidate entities. Then, min-max scaling is also applied to the logits to produce predicted probabilities $P = \{p_1, \ldots, p_K\}$, where p_i is the probability of option $o_i \in O$, i.e., the predicted probability of candidate e_{t_i} being the correct answer. Finally, the ranking loss $\mathcal{L}_{Rank}$ between the target ranking from first stage and the predicted ranking from LLM is:

$$\mathcal{L}_{Rank} = \frac{C}{K^2} \sum_{s_i^* < s_j^*} \max(0, p_i - p_j), i, j \in \{1, \ldots, K\} \quad (1)$$

where $C = 100$, and $\frac{C}{K^2}$ is the scaling term used to make $\mathcal{L}_{Rank}$ insensitive to K, since the number of additive terms in $\mathcal{L}_{Rank}$ is $K^2/2$. Specifically, the scaling term equals 1 when K is set to 10.

Training We fine-tune the LLM through the following objective:

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda \mathcal{L}_{Rank} \quad (2)$$

where weight $\lambda \in [0,1]$ and $\mathcal{L}_{CE}$ is the cross-entropy loss typically used in generative LLMs. To construct training samples consisting of input sequence x_{in} and target sequence y, we utilize the trained KGE model of the first stage to infer queries in the training set and get the predicted top-K candidates E_c as well as their corresponding scores S_c. The candidates in E_c are shuffled to form x_{in}, and y is obtained by sorting these candidates based on their scores. Each training triple (e_h, r, e_t) makes up two samples by separately querying $(e_h, r, ?)$ and $(?, r, e_t)$, namely queries that missing the tail and the head. No negative samples need to be constructed.

4.3. Knowledge-augmented Constrained Inference

Although the fine-tuned LLMs can to some extent capture the knowledge stored in KG and learn the output format, there exists omission issue where some or all of the candidates are not generated. The latter case typically arises from a lack of contextual knowledge, leading to a refusal to generate a ranking. To enhance the reasoning ability at inference stage, we retrieve two kinds of knowledge as prompts to assist the input side, namely query-related prompt and candidate-supporting prompt. Moreover, we design a constrained option generation method at the output side, to ensure that the answer exists and is validly ordered.

Query-related Prompt We retrieve query-related training triples to provide contextual knowledge for answering a query. Formally, each training triple $(e_h, r, e_t) \in \mathcal{F}$ is first converted into a triple sequence x_{hrt}, which is part of the training text set $\mathcal{X}_{train}$. This process is the same as obtaining the candidate triple sequence x_{hrt_i} described in Sec. 4.2. Similar to text retrieval methods, we embed the query sequence x_q and all training triples' texts $\mathcal{X}_{train}$ into semantic representation space using an off-the-shelf sentence embedding model, e.g., SBERT (Reimers and Gurevych, 2019). Then we calculate their cosine similarity and use only the top-K_q triple sequences in $\mathcal{X}_{train}$ similar to the given query x_q, which are concatenated into a query-related prompt x_q^k. Denote this process as F, which can be formulated as:

$$x_q^k = F(x_q, \mathcal{X}_{train}, K_q) \tag{3}$$

Candidate-supporting Prompt Taking a step further, we retrieve for each candidate the evidence supporting it as the answer from known training triples. To achieve this, we take the candidate triple sequence x_{hrt_i} obtained in *query-candidate interaction* as the query and retrieve its top-K_c similar triple sequences from $\mathcal{X}_{train}$ as support for candidate e_{t_i}, denoted as $x_{e_{t_i}}^k$, which is analogous to gaining query-related prompt:

$$x_{t_i}^k = F(x_{hrt_i}, \mathcal{X}_{train}, K_c) \tag{4}$$

Then we concatenate the supporting sequences of each candidate to form the candidate-supporting prompt x_c^k:

$$x_c^k = [x_{t_1}^k, x_{t_2}^k, \ldots, x_{t_K}^k] \tag{5}$$

where [] means concatenation of texts. To control the length of x_c^k and retain only the most useful supporting information, we additionally set a similarity threshold $\theta \in [0,1]$ during retrieval. Therefore, Equation (4) can be modified as:

$$x_{t_i}^k = F(x_{hrt_i}, \mathcal{X}_{train}, K_c, \theta) \tag{6}$$

Note that x_c^k could be empty text if none of the candidates has supporting information that satisfies the condition.

Constrained Option Generation During decoding, the LLM may suffer from the problem of being unable to output a complete ranking of option identifiers, even with the process of fine-tuning or the provision of context. Since the output format in this paper is simplified without generating labeled name texts of candidate entities, we propose to restrict the model to output all option identifiers without duplication, thus yielding a valid ranking. Due to the enormous permutation number of option identifiers, instead of using prefix constraints by enumerating all legal outputs (Chen et al., 2022), we directly narrow down the legitimate words from the entire vocabulary to the set of candidate option identifiers that have not appeared, at each position where an option identifier should be generated.

Inference Due to the model's instruction understanding capability, we employ a new instruction template T_{in}^k during inference to transform x_{in} into knowledge-augmented input sequence x_{in}^k, which contains query-related prompt x_q^k and candidate-supporting prompt x_c^k, noted as $x_{in}^k = T_{in}^k(x_q, x_c, x_q^k, x_c^k)$. Note that we do not use these two retrieved prompts during the training phase, as they may contain noise that could affect the learning process. The ordering of each option identifier in the generated text is used as the ranking of the corresponding candidate entity for evaluation.

5. Experimental Setup

5.1. Dataset

Following PKGC (Lv et al., 2022), we apply two curated KG datasets named Wiki27K and FB15K-

Dataset	$\|\mathcal{E}\|$	$\|\mathcal{R}\|$	# Train	# Valid	# Test
Wiki27K	27,122	62	74,793	10,121	10,122
FB15K-237-N	13,104	93	87,282	7,041	8,226
ReVerb20K	11,065	11,058	15,499	1,550	2,325
ReVerb45K	27,008	21,623	35,970	3,598	5,395

Table 1: Dataset statistics.

Dataset	Wiki27K	FB15K-237-N	ReVerb20K	ReVerb45K
K	20	20	30	30
λ	0.1	0.1	0.3	1.0

Table 2: Values of Hyperparameters.

237-N, which are sampled from Wikidata and Freebase. We also use two open KG datasets following CaRe (Gupta et al., 2019), namely ReVerb20K and ReVerb45K, which are extracted from text corpus by open information extraction approach ReVerb (Fader et al., 2011). Note that triples in open KG are in the form of **(head noun phrase, relation phrase, tail noun phrase)**, where the noun phrase (NP) and relation phrase (RP) are not canonicalized. This means that there exists NPs that link to the same entity, such as "Microsoft" and "Microsoft Corporation", and RPs that refer to the same relation, e.g., "be a close friend of" and "become good friend with". Gold canonical clusters of NPs extracted through Freebase entity linking information (Gabrilovich et al., 2013) are provided for evaluating missing tail NPs. The statistics of these datasets are listed in Table 1. For more details, we refer readers to the related papers.

5.2. Evaluation Metrics

We verify our approach on the link prediction task under filtered setting (Bordes et al., 2013) by standard ranking metrics: mean rank (MR), mean reciprocal rank (MRR), and hits at n (Hits@n), $n = \{1, 3, 10\}$. Note that in Open KG, we follow CaRe (Gupta et al., 2019) to evaluate the rank of canonical clusters for the target NP.

5.3. Experimental Settings

At the first ranking stage, for Wiki27K and FB15K-237-N, we apply TuckER (Balazevic et al., 2019) as the KGE model and set the embedding dimension $d = 256$ following PKGC (Lv et al., 2022). For ReVerb20K and ReVerb45K, we follow CEKFA (Wang et al., 2023b) and employ its KGE model, noted as CEKFA-KFARe, where $d = 768$.

During the second stage, we freeze the parameters of the KGE model and perform inference on the training set to obtain training samples for KC-GenRe. The instruction templates used in experiments are listed in Table 3. Please note that we do not consider designing alternative instructions, as this is not the focus of this paper. We use manually

T_{in}	Below is an instruction that describes a task, paired with a question and corresponding candidate answers. The questions and candidate answers have been combined into candidate corresponding statements. Combine what you know, output a ranking of these candidate answers.\n\n ### Question: $\{x_q\}$\n\n $\{x_c\}$ ### Response:
T_{in}^k	Below is an instruction that describes a task, paired with a question and corresponding candidate answers. The questions and candidate answers have been combined into candidate corresponding statements. Knowledge related to some candidates will be provided that may be useful for ranking. Combine what you know and the following knowledge, output a ranking of these candidate answers.\n\n ### Supporting information: $\{x_q^k\}$\n\n ### Candidate supporting knowledge: $\{x_c^k\}$\n\n ### Question: $\{x_q\}$\n\n $\{x_c\}$ ### Response:

Table 3: Instruction templates of KC-GenRe, where x_q, x_c, x_q^k and x_c^k represent query sequence, candidate sequence, query-related prompt, and candidate-supporting prompt respectively.

constructed relational prompts following PKGC (Lv et al., 2022) to convert triples into natural sentences on Wiki27K and FB15K-237-N datasets, so as to obtain x_q, x_{hrt_i} and x_{hrt}. While on ReVerb20K and ReVerb45K, we directly concatenate NPs and RPs in a triple into sentence, given that they are expressed in natural language.

Built on LLaMA-7b (Touvron et al., 2023a), KC-GenRe is fine-tuned by QLORA (Dettmers et al., 2023) approach. The re-ranking number K is chosen from {10,20,30}, and its value remains the same during both training and testing, unless specifically stated. Loss weight λ is picked from {0, 0.1, 0.3, 0.5, 0.7, 0.9, 1.0}. The best settings of hyperparameters are listed in Table 2. KC-GenRe is fine-tuned for a maximum of 3 epochs, with a batch size of 16 and a learning rate of $1e$-4 on all datasets. Same as PKGC (Lv et al., 2022), we adopt entity definition as auxiliary knowledge during training on Wiki27K and FB15K-237-N. As a result, during inference, we do not additionally use query-related prompt and candidate-supporting prompt as contextual knowledge on these two datasets. While on ReVerb20K and ReVerb45K, the query-related prompt and candidate-supporting prompt are both utilized to assist in the inference process, and the values of K_q, K_c, θ are empirically set to 3, 3, 0.8 without tuning. We employ the pre-trained all-mpnet-base-v2 SBERT model (Reimers and Gurevych, 2019) to encode texts into sentence embeddings for retrieval in Eq.(3) and Eq.(6). Greedy search decoding strategy is applied. All experiments are conducted in Pytorch and on one 80G A800 GPU.

Model	Wiki27K				FB15K-237-N			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
TransE† (Bordes et al., 2013)	0.155	0.032	0.228	0.378	0.255	0.152	0.301	0.459
TransC† (Lv et al., 2018)	0.175	0.124	0.215	0.339	0.233	0.129	0.298	0.395
ConvE† (Dettmers et al., 2018)	0.226	0.164	0.244	0.354	0.273	0.192	0.305	0.429
WWV† (Veira et al., 2019)	0.198	0.157	0.237	0.365	0.269	0.137	0.287	0.443
TuckER (Balazevic et al., 2019)	0.249	0.185	0.269	0.385	0.309	0.227	0.340	0.474
RotatE† (Sun et al., 2019)	0.216	0.123	0.256	0.394	0.279	0.177	0.320	0.481
KG-BERT† (Yao et al., 2019)	0.192	0.119	0.219	0.352	0.203	0.139	0.201	0.403
LP-RP-RR† (Kim et al., 2020)	0.217	0.138	0.235	0.379	0.248	0.155	0.256	0.436
PKGC† (Lv et al., 2022)	0.285	0.230	0.305	**0.409**	0.332	0.261	0.346	0.487
KC-GenRe	**0.317**	**0.274**	**0.330**	0.408	**0.399**	**0.338**	**0.427**	**0.505**

Table 4: Link prediction results on two curated KGs. Best results are in bold and second best are underlined. [†]: results are taken from PKGC (Lv et al., 2022).

Model	ReVerb20K					ReVerb45K				
	MRR	MR	Hits@1	Hits@3	Hits@10	MRR	MR	Hits@1	Hits@3	Hits@10
TransE (Bordes et al., 2013)	0.138	1150.5	0.034	0.201	0.316	0.202	1889.5	0.122	0.243	0.346
ComplEx (Trouillon et al., 2016)	0.038	4486.5	0.017	0.043	0.071	0.068	5659.8	0.054	0.071	0.093
R-GCN (Schlichtkrull et al., 2018)	0.122	1204.3	-	-	0.187	0.042	2866.8	-	-	0.046
ConvE (Dettmers et al., 2018)	0.262	1483.7	0.203	0.287	0.371	0.218	3306.8	0.166	0.243	0.314
KG-BERT (Yao et al., 2019)	0.047	420.4	0.014	0.039	0.105	0.123	1325.8	0.070	0.131	0.223
RotatE (Sun et al., 2019)	0.065	2861.5	0.043	0.069	0.108	0.141	3033.4	0.110	0.147	0.196
PairRE (Chao et al., 2021)	0.213	1366.2	0.166	0.229	0.296	0.205	2608.4	0.153	0.228	0.302
ResNet (Lovelace et al., 2021)	0.224	2258.4	0.188	0.240	0.292	0.181	3928.9	0.150	0.196	0.242
BertResNet-ReRank (Lovelace et al., 2021)	0.272	1245.6	0.225	0.294	0.347	0.208	2773.4	0.166	0.227	0.281
CaRe (Gupta et al., 2019)	0.318	973.2	-	-	0.439	0.324	1308.0	-	-	0.456
OKGIT (Chandrahas and Talukdar, 2021)	0.359	527.1	0.282	0.394	0.499	0.332	773.9	0.261	0.363	0.464
OKGSE (Xie et al., 2022a)	0.372	487.3	0.291	0.408	0.524	0.342	**771.1**	0.274	0.371	0.473
CEKFA (Wang et al., 2023b)	0.387	416.7	0.310	0.427	0.515	0.369	884.5	0.294	0.409	0.502
KC-GenRe	**0.408**	**410.8**	**0.331**	**0.450**	**0.547**	**0.404**	874.1	**0.332**	**0.444**	**0.534**

Table 5: Link prediction results on two open KGs. Best results are in bold and second best are underlined.

5.4. Baselines

For curated KG datasets Wiki27K and FB15K-237-N, we adopt several models for comparison, including (1) KGE-based methods: TransE (Bordes et al., 2013), ConvE (Dettmers et al., 2018), TuckER (Balazevic et al., 2019), RotatE (Sun et al., 2019), TransC (Lv et al., 2018), WWV (Veira et al., 2019), where the last two use concept and definition information, respectively. (2) LM-based methods: KG-BERT (Yao et al., 2019), LP-RP-RR (Kim et al., 2020), and PKGC (Lv et al., 2022), where the last one uses entity definition information.

For open KG datasets ReVerb20K and ReVerb45K, the comparison include: (1) methods in curated KG: TransE (Bordes et al., 2013), ComplEx (Trouillon et al., 2016), R-GCN (Schlichtkrull et al., 2018), ConvE (Dettmers et al., 2018), KG-BERT (Yao et al., 2019), RotatE (Sun et al., 2019), PairRE (Chao et al., 2021), the two-stage method (Lovelace et al., 2021), notated as BertResNet-ReRank, and the query encoding module in it, notated as ResNet; (2) methods in Open KG: CaRe (Gupta et al., 2019), OKGIT (Chandrahas and Talukdar, 2021), OKGSE (Xie et al., 2022a) and CEKFA (Wang et al., 2023b). Among them, KG-BERT, OKGIT, OKGSE, BertResNet-ReRank, and CEKFA are LM-based methods, with the latter two employing a two-stage reranking architecture.

6. Experimental Results

6.1. Main Results

By targeting the re-ranking of candidate option identifiers and performing constrained option generation, KC-GenRe thoroughly resolves the issues of mismatch and omission in generative KGC based on LLMs. While the misordering problem can be assessed through the metrics presented in Table 4 and Table 5. It can be seen that KC-GenRe outperforms existing works on both curated and open KGs. In Table 4, we gain absolute improvements of 3.2% and 6.7% for MRR, 4.4% and 7.7% for Hits@1 on Wiki27K and FB15K-237-N compared to PKGC, which owns the same KGE ranking model (TuckER) as ours. Compared to TuckER, our re-ranking method KC-GenRe obatins increases of 6.8% and 9.0% for MRR, 8.9% and 11.1% for Hits@1. In Table 5, KC-GenRe achieves higher performance than previous methods with improvements of 2.1% and 3.5% for MRR, 2.1% and 3.8% for Hits@1 on ReVerb20K and ReVerb45K, respectively. The sub-optimal MR on the ReVerb45K may stem from a few poor predictions (e.g., predicted rankings exceeding 10), thus enlarging the mean number of rankings. Nevertheless, KC-GenRe achieves superior performance in terms of MRR and Hits@1, which are considered more reliable and accurate.

From the tables, we can learn that the LM-only approach may not be comparable to traditional KGE approaches. For example, the Hits@1 metrics implemented by KG-BERT (Yao et al., 2019) are lower than Tucker (Balazevic et al., 2019) by 6.4% and 8.9% on Wiki27K and FB15K-237-N, and are lower than ConvE (Dettmers et al., 2018) by 18.9% and 9.6% on ReVerb20K and ReVerb45K, respectively. By combining KGE and LM to perform coarse ranking and fine-grained re-ranking respectively, it is possible to ensure high efficiency and recall in the ranking stage, and to obtain further accurate predictions through LMs.

6.2. Ablation Study

Table 6 and Table 7 show impacts of each component proposed in KC-GenRe, namely query-candidate interaction (QCI), candidate-candidate interaction (CCI), query-related prompt (QP), candidate-supporting prompt (CP), and constrained option generation (CG). The first field (Base) in these tables represents the baseline KGE model without re-ranking, and the last two fields are re-ranking models based on generative LLMs. Specifically, entry "1" indicates the re-ranking method that directly fine-tunes the LLMs without using all proposed components.

It can be found that removing individual component or their combinations can lead to performance decreases. And the most significant improvements can be observed when all components are adopted, with gains of 4.1%, 5.2%, 6.8% and 9.0% in MRR on ReVerb20K, ReVerb45K, Wiki27K and FB15K-237-N respectively, compared to the KGE model without re-ranking (Base). And the Hits@1 metric achieves the most significant improvement on FB15K-237-N by 11.1%.

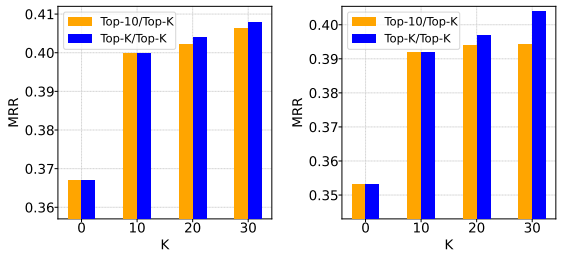
Concretely, (1) Directly fine-tuning the generative LLMs for re-ranking without using any components of KC-GenRe (entry 1) can still be effective on Wiki27K and FB15K-237-N. However, it brings no performance improvement or even a drop of 1.7% and 1.8% in MRR on ReVerb20K and ReVerb45K, compared to the KGE model (Base). This illustrates the necessity of exploring methods based on LLM that are applicable to KGC task. (2) The query-candidate interaction is clearly critical for KC-GenRe, with an increased MRR of 2.5% and 3.9% on ReVerb20K and ReVerb45K respectively, compared to simply listing all candidates (entry 2). It indicates that combining each candidate with the query can fully learn the possibility of each candidate completing the query. (3) The ranking loss calculated by the candidate-candidate interaction (entry 3) is also helpful. For instance, on FB15K-237-N, it makes increases in MRR and Hits@1 metrics of 3.8% and 5.4% respectively, illustrating the usefulness of learning relative order between candidates. (4) In Table 6, by utilizing both query-related prompt and candidate-supporting prompt together (entry 6) in inference, KC-GenRe achieves 3.8% and 4.4% MRR boosts. This implies that the retrieved relevant training triples indeed provide useful contextual knowledge so that the reasoning ability of LLM can be fully exploited. When removing one of these two prompts (entry 4 and 5), the performance of KC-GenRe decreases slightly, indicating that they may contain overlapping extracted triplets. (5) In Table 7, the removal of entity definition prompt (entry 4) has a notable effect on both datasets, emphasizing the importance of providing contextual knowledge for reasoning. (6) It is worth noting that the performance drop is pronounced when only constrained option generation is removed (entry 7 and entry 5 in two tables), even may be slightly worse than entry "Base". We analyze the experimental results and find that it is better to use constrained option generation together with candidate-candidate interaction. This could be due to the fact that if we utilize the logits of option identifiers to calculate the candidate-candidate interaction loss and influence the probability learning of each option identifier during training, a corresponding decoding using the logits of these option identifiers (i.e., constrained option generation) is required during inference. Failure to do so may disrupt the original decoding balance of LLMs and result in decreased performance. In conclusion, these results highlight the effectiveness of KC-GenRe in enhancing KGC re-ranking based on generative LLMs through the utilization of the proposed components.

	QCI	CCI	QP	CP	CG	ReVerb20K		ReVerb45K	
						MRR	Hits@1	MRR	Hits@1
Base						0.367	0.288	0.352	0.274
1						0.350	0.263	0.334	0.246
2		✓	✓	✓	✓	0.383	0.304	0.365	0.282
3	✓		✓	✓	✓	0.405	0.326	0.381	0.298
4	✓	✓		✓	✓	0.399	0.320	0.397	0.323
5	✓	✓	✓		✓	0.403	0.328	0.400	0.327
6	✓	✓			✓	0.370	0.284	0.360	0.271
7	✓	✓	✓	✓		0.367	0.289	0.352	0.273
KC-GenRe	✓	✓	✓	✓	✓	0.408	0.331	0.404	0.332

Table 6: Ablation results on open KGs, where "Base" is CEKFA-KFARe (Wang et al., 2023b).

Model	QCI	CCI	DP	CG	Wiki27K		FB15K-237-N	
					MRR	Hits@1	MRR	Hits@1
Base					0.249	0.185	0.309	0.227
1					0.283	0.227	0.329	0.248
2		✓	✓	✓	0.314	0.268	0.340	0.257
3	✓		✓	✓	0.311	0.266	0.361	0.284
4	✓	✓		✓	0.298	0.249	0.353	0.277
5	✓	✓	✓		0.245	0.178	0.303	0.218
KC-GenRe	✓	✓	✓	✓	0.317	0.274	0.399	0.338

Table 7: Ablation results on curated KGs, where "Base" denotes TuckER (Balazevic et al., 2019) and "DP" represents entity definition prompt.



(a) ReVerb20K (b) ReVerb45K

Figure 3: Effects of re-ranking number K. The front and back of the slash in the legend represent the values of K during training and testing, respectively.

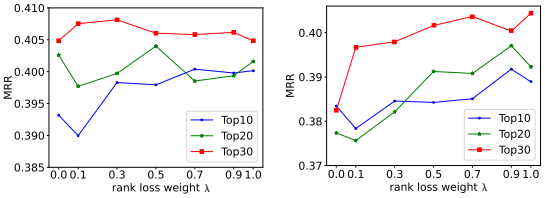
6.3. Influences of Re-ranking Number

To investigate the effects of the re-ranking number K, we conduct experiments in two cases: (1) the number of candidates for re-ranking during inference is larger than that during training; (2) the number of candidates for re-ranking during inference is the same as that during training. For case (1), we train the model with top-10 (K=10) candidates and re-rank top-K at inference, while for case (2), we train and test the model with top-K candidates.

Figure 3 shows the results for case (1) (orange) and case (2) (blue). It is obvious that increasing the re-ranking number in both cases leads to performance improvements on both datasets. This is mainly due to the increase of samples containing correct answers in candidates. In addition, the results of orange bars indicate that the model trained to rank the top-K candidates has the ability to rank beyond the top-K. However, in Figure 3(b), they rise slightly with K increased and clearly show poorer performance than blue bars. It is reasonable because training on more candidates, i.e., negative answers, allows for the learning of more knowledge and the achievement of higher performance. This is also the reason why the results of case (2) is better than that of case (1).

6.4. Effects of Candidate-candidate Interaction

We compare the influences of weight λ for ranking loss $\mathcal{L}_{Rank}$ in candidate-candidate interaction when training and re-ranking with different top-K. As shown in Figure 4, the prediction performance is able to be improved after applying candidate-candidate interaction. Although the distributions on these datasets are not the same, we can learn that the overall performance is generally better as K gets larger. Besides, we find that performance drops when λ is small ($\lambda = 0.1$) with $K \leq 20$, while it boosts with $K > 20$. This may be due to the fact that sorting difficulty rises with the increase of re-ranking number K, leading to a larger ranking loss



(a) ReVerb20K (b) ReVerb45K

Figure 4: Influences of weight λ in Eq.(2) with different re-ranking number K (Top-K).

LLM	ReVerb20K		ReVerb45K	
	MRR	Hits@1	MRR	Hits@1
LLaMA-7b	0.400	0.324	0.392	0.325
LLaMA2-7b	0.406	0.334	0.397	0.329
LLaMA-13b	0.403	0.331	0.392	0.322
LLaMA-65b	0.400	0.326	0.404	0.342

Table 8: Link prediction results of KC-GenRe implemented with different LLMs when $K = 10$.

$\mathcal{L}_{Rank}$, so that a smaller value of λ could balance it with $\mathcal{L}_{CE}$ and get commendable performance.

6.5. Impacts with Different LLMs

We compare the results of KC-GenRe using different LLMs, as shown in Table 8. Various LLMs exhibit varying levels of performance, e.g., LLaMA2-7b (Touvron et al., 2023b) achieves higher Hits@1 than LLaMA-7b on ReVerb20K and ReVerb45K by 1.0% and 0.4%, respectively. Increasing the scale of LLMs may bring gains, such as LLaMA-65b gets a 1.7% Hits@1 increase on ReVerb45K over LLaMA-7b. However, the lift may be slight. This could be due to the limited or unmemorized increase of knowledge directly related to the query in the corpus used for pre-training the LLM, resulting in no remarkable improvement in the ability to answer factual questions. Additionally, a 7B model may be sufficient to achieve optimized results in KGC task (Yang et al., 2023a).

7. Conclusion

This paper introduces KC-GenRe, a knowledge-constrained generative re-ranking model for KGC. To tackle mismatch, misordering, and omission issue, we formulates the task as a candidate identifier sorting generation problem and design a knowledge-guided interactive training method as well as a knowledge-augmented constrained inference method. KC-GenRe can enhance the identification and relative ranking of candidates, and generate valid results with supporting prompts. Experimental results demonstrate its superior performance and highlighting its effectiveness for KGC.

8. Acknowledgements

This work has been partly supported by the National Natural Science Foundation of China (Grant No. 62025208 and Grant No. 62376284), and the Xiangjiang Laboratory Foundation (Grant No. 22XJ01012).

9. Bibliographical References

Ivana Balazevic, Carl Allen, and Timothy Hospedales. 2019. TuckER: Tensor factorization for knowledge graph completion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 5185–5194. Association for Computational Linguistics.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of the Advances in Neural Information Processing Systems 26: Annual Conference on Neural Information Processing Systems*, NIPS'13, page 2787–2795. Curran Associates Inc.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779. 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 *Proceedings of the Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems*.

. Chandrahas and Partha Talukdar. 2021. OKGIT: Open knowledge graph link prediction with implicit types. In *Proceedings of the Findings of the Association for Computational Linguistics*, pages 2546–2559. Association for Computational Linguistics.

Linlin Chao, Jianshan He, Taifeng Wang, and Wei Chu. 2021. PairRE: Knowledge graph embeddings via paired relation vectors. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 4360–4369. Association for Computational Linguistics.

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. International Committee on Computational Linguistics.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence and 30th Innovative Applications of Artificial Intelligence Conference and 8th AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18, pages 1811–1818. AAAI Press.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

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*, pages 4171–4186. Association for Computational Linguistics.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545.

Evgeniy Gabrilovich, Michael Ringgaard, and Amarnag Subramanya. 2013. Facc1: Freebase annotation of clueweb corpora, version 1 (release date 2013-06-26, format version 1, correction level 0). *Note: http://lemurproject.org/clueweb09/FACC1/Cited by*, 5:140.

Swapnil Gupta, Sreyash Kenkre, and Partha Talukdar. 2019. Care: Open knowledge graph embeddings. In *Proceedings of the 2019 Conference on*

Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 378–388.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *Proceedings of the 10th International Conference on Learning Representations*.

Pengcheng Jiang, Shivam Agarwal, Bowen Jin, Xuan Wang, Jimeng Sun, and Jiawei Han. 2023. Text-augmented open knowledge graph completion via pre-trained language models. In *Proceedings of the Findings of the Association for Computational Linguistics*, pages 11161–11180. 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.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Proceedings of the Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems*, 35:22199–22213.

Keshav Kolluru, Mayank Singh Chauhan, Yatin Nandwani, Parag Singla, et al. 2021. Cear: Cross-entity aware reranker for knowledge base completion. *arXiv preprint arXiv:2104.08741*.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 61–68.

Justin Lovelace, Denis Newman-Griffis, Shikhar Vashishth, Jill Fain Lehman, and Carolyn P. Rosé. 2021. Robust knowledge graph completion with stacked convolutions and a student re-ranking network. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 1016–1029. Association for Computational Linguistics.

Xin Lv, Lei Hou, Juanzi Li, and Zhiyuan Liu. 2018. Differentiating concepts and instances for knowledge graph embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1971–1979. Association for Computational Linguistics.

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 *Proceedings of the Findings of the Association for Computational Linguistics*, pages 3570–3581. Association for Computational Linguistics.

Xueguang Ma, Xinyu Zhang, Ronak Pradeep, and Jimmy Lin. 2023. Zero-shot listwise document reranking with a large language model. *arXiv preprint arXiv:2305.02156*.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064. Association for Computational Linguistics.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Proceedings of the Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems*, 35:27730–27744.

Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023. Rankvicuna: Zero-shot listwise document reranking with open-source large language models. *arXiv preprint arXiv:2309.15088*.

Wei Qian, Cong Fu, Yu Zhu, Deng Cai, and Xiaofei He. 2018. Translating embeddings for knowledge graph completion with relation attention mechanism. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4286–4292. International Joint Conferences on Artificial Intelligence Organization.

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. *Journal of Machine Learning Research*, 21(1).

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3982–3992. Association for Computational Linguistics.

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*, pages 2814–2828. Association for Computational Linguistics.

Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. 2020. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4498–4507. Association for Computational Linguistics.

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *Proceedings of the 15th Extended Semantic Web Conference*, pages 593–607. Springer International Publishing.

Yueqing Sun, Qi Shi, Le Qi, and Yu Zhang. 2022. JointLK: Joint reasoning with language models and knowledge graphs for commonsense question answering. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5049–5060. Association for 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 *Proceedings of the 7th International Conference on Learning Representations*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

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. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

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

Neil Veira, Brian Keng, Kanchana Padmanabhan, and Andreas Veneris. 2019. Unsupervised embedding enhancements of knowledge graphs using textual associations. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 5218–5225. International Joint Conferences on Artificial Intelligence Organization.

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

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*, pages 4281–4294. Association for Computational Linguistics.

Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, et al. 2023a. Instructuie: Multi-task instruction tuning for unified information extraction. *arXiv preprint arXiv:2304.08085*.

Yilin Wang, Minghao Hu, Zhen Huang, Dongsheng Li, Wei Luo, Dong Yang, and Xicheng Lu. 2023b. A canonicalization-enhanced known fact-aware framework for open knowledge graph link prediction. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 2332–2342. International Joint Conferences on Artificial Intelligence Organization.

Robert West, Evgeniy Gabrilovich, Kevin Murphy, Shaohua Sun, Rahul Gupta, and Dekang Lin. 2014. Knowledge base completion via search-based question answering. In *Proceedings of*

the 23rd International Conference on World Wide Web, WWW '14, page 515–526, New York, NY, USA. Association for Computing Machinery.

Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. 2016a. Representation learning of knowledge graphs with entity descriptions. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, AAAI'16, page 2659–2665. AAAI Press.

Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2016b. Representation learning of knowledge graphs with hierarchical types. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, IJCAI'16, page 2965–2971. AAAI Press.

Tingyu Xie, Peng Peng, Hongwei Wang, and Yusheng Liu. 2022a. Open knowledge graph link prediction with segmented embedding. In *Proceedings of the 2022 International Joint Conference on Neural Networks*, pages 1–8.

Xin Xie, Ningyu Zhang, Zhoubo Li, Shumin Deng, Hui Chen, Feiyu Xiong, Mosha Chen, and Huajun Chen. 2022b. From discrimination to generation: Knowledge graph completion with generative transformer. In *Companion Proceedings of the Web Conference*, WWW '22, page 162–165. Association for Computing Machinery.

Rui Yang, Li Fang, and Yi Zhou. 2023a. Can text-based knowledge graph completion benefit from zero-shot large language models? *arXiv preprint arXiv:2310.08279*.

Yuhao Yang, Chao Huang, Lianghao Xia, and Chenliang Li. 2022. Knowledge graph contrastive learning for recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 1434–1443. Association for Computing Machinery.

Zonglin Yang, Xinya Du, Erik Cambria, and Claire Cardie. 2023b. End-to-end case-based reasoning for commonsense knowledge base completion. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3491–3504. Association for Computational Linguistics.

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

Liang Yao, Jiazhen Peng, Chengsheng Mao, and Yuan Luo. 2023. Exploring large language models for knowledge graph completion. *arXiv preprint arXiv:2308.13916*.

Zhiyuan Zhang, Xiaoqian Liu, Yi Zhang, Qi Su, Xu Sun, and Bin He. 2020. Pretrain-KGE: Learning knowledge representation from pretrained language models. In *Proceedings of the Findings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 259–266. Association for Computational Linguistics.

Yuqi Zhu, Xiaohan Wang, Jing Chen, Shuofei Qiao, Yixin Ou, Yunzhi Yao, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2023a. Llms for knowledge graph construction and reasoning: Recent capabilities and future opportunities. *arXiv preprint arXiv:2305.13168*.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Zhicheng Dou, and Ji-Rong Wen. 2023b. Large language models for information retrieval: A survey. *arXiv preprint arXiv:2308.07107*.