Contextual Knowledge Graph

Chengjin Xu[1*], Muzhi Li[1,2*] , Cehao Yang[1], Xuhui Jiang[1,3], Lumingyuan Tang[1], Yiyan Qi[1], Jian Guo[1†]

1. IDEA Research, International Digital Economy Academy
2. Department of Computer Science and Engineering, The Chinese University of Hong Kong
3. CAS Key Laboratory of AI Safety, Institute of Computing Technology, CAS
{xuchengjin,limuzhi,yangcehao,jiangxuhui, guojian}@idea.edu.cn

Abstract

Knowledge Graphs (KGs) are foundational structures in many AI applications, representing entities and their interrelations through triples. However, triple-based KGs lack the contextual information of relational knowledge, like temporal dynamics and provenance details, which are crucial for comprehensive knowledge representation and effective reasoning. Instead, **Contextual Knowledge Graphs** (CKGs) expand upon the conventional structure by incorporating additional information such as time validity, geographic location, and source provenance. This integration provides a more nuanced and accurate understanding of knowledge, enabling KGs to offer richer insights and support more sophisticated reasoning processes. In this work, we first discuss the inherent limitations of triple-based KGs and introduce the concept of contextual KGs, highlighting their advantages in knowledge representation and reasoning. We then present **KGR[3], a context-enriched KG reasoning paradigm** that leverages large language models (LLMs) to retrieve candidate entities and related contexts, rank them based on the retrieved information, and reason whether sufficient information has been obtained to answer a query. Our experimental results demonstrate that KGR[3] significantly improves performance on KG completion (KGC) and KG question answering (KGQA) tasks, validating the effectiveness of incorporating contextual information on KG representation and reasoning.

1 Introduction

Knowledge Graphs (KGs) are structured knowledge bases (KBs) that organize factual knowledge as triples in the form of *(head entity, relation, tail entity)*. These triples interweave into a graph-like structure, where each node represents an entity and each edge represents a relationship. This structured representation enables machines to easily understand and reason about knowledge, thereby supporting various intelligent applications such as question answering (Sun et al., 2024), semantic analysis (Wang and Shu, 2023), recommendation systems (Wang et al., 2019), and more.

While this triple-based structure offers clear semantics and precision through the use of schemas and ontologies, it loses the contextual information of knowledge and falls short in capturing the complexity and richness of real-world knowledge (Dong, 2023). Since we cannot clearly model the knowledge in a domain only with entities and relations, many recent KGs (Pellissier Tanon et al., 2020; Tharani, 2021) are designed to be semi-structured: they leverage the clear semantics of structured data provided by the rigidity of schemas (i.e., ontologies) while also embracing the flexibility of unstructured data. Such KGs integrate multi-modal knowledge, including entity description, images, timestamps and other metadata, all of which can be regarded as the contexts of triple knowledge. In this paper, we refer to this type of KGs as contextual KGs (CKGs). By incorporating these semantic contexts, contextual KGs provide a more comprehensive and nuanced representation of knowledge, extending beyond the traditional triple-based approach. This enables KGs to possess more advanced capabilities in knowledge representation and reasoning.

Moreover, large language models (LLMs), pre-trained on vast text corpora, have exhibited strong semantic understanding capability (Brown et al., 2020a). And the use of LLMs for KG reasoning has become a research hotspot (Wei et al., 2023; Liao et al., 2024; Sun et al., 2024). However, KGs may contain numerous entities and relations, but not all entities and relationships are fully annotated and connected, leading to data sparsity. This sparsity results in a lack of sufficient contextual infor-

---

*Both authors contributed equally to this research.
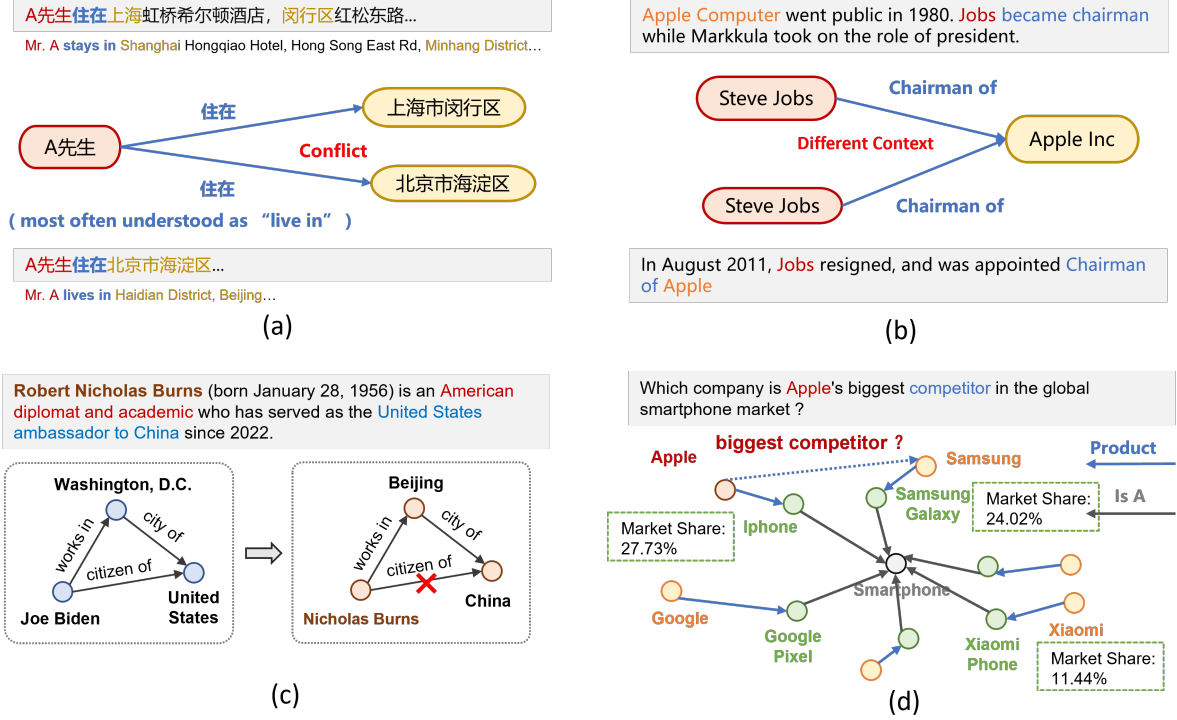†Corresponding author.

Figure 1: Examples of limitations of triple-based KGs. (a) gives an example that the loss of contextual information during KG construction processes may result in the extraction of contradictory triples; (b) gives an example that triple-based representation struggle to represent two facts that involve the same entities and relations but occur in different contexts; (c) gives an example that triple-based KG reasoning methods often learn rule patterns that frequently occur in KGs, but they tend to ignore contexts that may affect the validity of these rules; (d) gives an example that triple-based KG reasoning methods face difficulties in answering questions that involve relational knowledge or contextual information beyond the scope of the triples in KGs.

mation for the LLM during inference. On the other hand, LLMs are better at handling unstructured data rather than structured triples. Considering that contextual KGs can provide unstructured contextual information for LLM reasoning, the synergy between contextual KGs and LLMs holds significant potential for advancing the field of knowledge reasoning.

In this paper, we will first give a brief discussion on the limitations of the triple-based KGs and give the specific definition of contextual KGs. To validate the effectiveness of contextual KG on enhancing KG representation and reasoning, we propose a novel context-enhanced KG reasoning paradigm, named **KGR^3**, which leverage the strong reasoning power of LLMs to firstly **retrieve** candidate entities and related contexts from KG, and **rank** the candidate entities based on retrieved context, and then **reason** whether sufficient information is retrieved to answer the question. Experimental results demonstrate that our proposed paradigm KGR^3 enhances the performance of existing models on the tasks of KG completion (KGC) and KG question answering (KGQA), which are two of fun-

damental reasoning tasks over KGs. Experimental Overall, this paper have two major contributions:

- Point out the limitations of the current triple-based KGs, and introduce the concept of contextual KG, which has more advanced capabilities in knowledge representation and reasoning.

- Propose a context-enhanced KG reasoning paradigm, KGR^3, which leverages the LLM to perform KG reasoning based on related contexts. Experimental results on KGC and KGQA support our intuition that the integration of contextual data can contribute to effective KG reasoning.

2 Contextual Knowledge Graph

In this section, we first discuss on the limitations of triple-based KGs, caused by the absence of contextual information. Moreover, we point out the effects of contextual information on knowledge representation and reasoning, then categorize and interpreter different types of contexts in the KGs.

Finally, we formally define contextual KGs as well as two reasoning tasks over contextual KGs, i.e. KG completion (KGC) and KG question answering (KGQA).

2.1 Limitations of Triple-based KGs

A Triple-based Knowledge Graph (denoted as $\mathcal{KG} = \{\mathcal{E}, \mathcal{R}, \mathcal{T}\}$) can be represented as a set of triples in the form of $(h, r, t) \in \mathcal{T}$, where $h, t \in \mathcal{E}$, $r \in \mathcal{R}$. The notations h and t denote the head and the tail entity of a triple. $\mathcal{E}, \mathcal{R}, \mathcal{T}$ are the set of entities, relations, and triples, respectively. Typical triple-based KGs include Freebase (Bollacker et al., 2008), WordNet (Miller, 1995) and DBPedia (Lehmann et al., 2014). In these triple-based KGs, the triple representation excludes crucial contextual information, often resulting in inaccurate knowledge storage, incomplete representation, and ineffective reasoning. These issues are the primary constraints on the practical application of most current KGs.

To be specific, the same relationship may have different meanings in different contexts, thus the triple representation could lead to incorrect knowledge storage. For instance, consider the two sentences: 'A先生住在上海虹桥希尔顿酒店，闵行区红松东路' and 'A先生住在北京市海淀区' as shown in Figure 1(a). They may be represented as two triples: *(A先生, 住在, 上海市闵行区)* and *(A先生, 住在, 北京市海淀区)*, respectively, in a KG. However, these representations are semantically conflicting since a person cannot live in two places simultaneously. This mistake is likely to occur because the predicate '住' in the first sentence implies '*stay in*', whereas in the latter one, it denotes '*live in*'. The triple extraction process filters out the sentence context, leading to information conflicts.

Moreover, each data instance in a KG strictly adheres to its ontology structure. The ontology structure defines the categories of entities, relations, and attributes, as well as their hierarchical relationships. During the construction of a KG, knowledge outside the pre-defined categories is filtered out, including a large amount of contextual information, leading to incomplete knowledge representation. For example, the contexts of Steve Jobs serving as the chairman of Apple Inc. twice are very different as shown in Figure 1(b). However, based on triple representation, both events would be represented as (*Steve Jobs, chairman of, Apple Inc.*), which results in downstream tasks not obtaining sufficient

information when utilizing related knowledge.

Triple-based knowledge representation also limits the effectiveness of existing KG reasoning methods, which mainly focus on learning explicit or implicit rules through rule mining or embedding models. For example, from triple (*X, works in, Y*) and (*Y, city of, Z*), it is very likely for KG reasoning models to deduce that (*X, citizen of, Z*) since such rule pattern appears frequently in the training data as shown in Figure 1(c). However, these probability based rules may not hold in all contexts, leading to conclusions that do not align with the facts. Besides, triple-based KGs only contain relational knowledge limited by predefined relation set $\mathcal{R}$. The triple-based reasoning process have difficulties in answering questions involving relations out of $\mathcal{R}$ without additional contextual information or external data sources.

2.2 The Effects of Contextual Information

To address the limitations of triple-based KGs, a promising approach is to attach contextual data to factual triples. For instance, several KGs, such as YAGO and the Yahoo Knowledge Graph, include meta-information with their facts, such as the time of validity, the geographic location of a fact, and provenance information. By integrating such data, contextual KGs can offer a more comprehensive and accurate representation of knowledge, thereby enabling more effective reasoning.

Knowledge Representation: Contextual data provide additional layers of information that enhance the representation and understanding of facts. For example, contextual data can differentiate facts that have the same relations and entities but occur in different backgrounds, such as recurring events in history. This differentiation allows for a more nuanced and detailed understanding of the information, capturing the various dimensions in which similar facts can differ based on time, location, and other contextual elements.

Knowledge Reasoning: During the process of knowledge reasoning, contextual information within contextual KGs can be leveraged to associate entities that are not directly connected by identifying similar contexts. This capability is particularly useful for making connections and drawing inferences that go beyond the predefined relation set of a triple-based KG. Moreover, contextual information provides additional knowledge, allowing
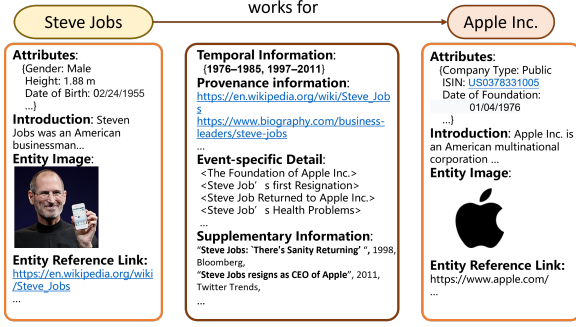
Figure 2: An example of factual triples with entity and relation contexts

for larger knowledge coverage and greater flexibility compared to triple-based KGs. Specifically, contextual information can be used to answer complex reasoning questions, such as those involving qualifiers or specific conditions that are often hidden within contextual data. For instance, answering a question about "*which company is Apple's biggest competitor in the global smartphone market*" would require integrating quantitative data, temporal information, and detailed market dynamics analysis with basic entity and relation information in KGs, as shown in Figure 1(d). Contextual KGs thus enable the handling of such intricate queries by providing a richer and more detailed knowledge base.

2.3 Categories of Contextual Data in KGs

As shown in Figure 2, contextual data in KGs can be roughly classified into two categories, i.e., entity contexts and triple contexts.

Entity contexts refer to information that provides a deeper understanding of an individual entity within the KG. This type of context helps in defining the attributes, characteristics and backgrounds of the entity. Entity contexts include entity attributes, entity types, entity descriptions, entity aliases, entity reference links, entity images, entity speeches, entity videos, etc.

Relation contexts refer to specific pieces of information that describe the relations between entities. They provides concrete data points and factual statements that contribute to the KG's informational content. Relation contexts include temporal information, geographic locations, quantitative data, provenance information, confidence levels, event-specific details, and other supplementary information. By incorporating these relation contexts, KGs can offer a richer, more detailed representa-

tion of the relationships between entities, enhancing their overall accuracy and utility for reasoning and analysis.

Table 1 demonstrates some examples of different types of entity contexts and relation contexts.

2.4 Problem Specification

A Contextual Knowledge Graph (denoted as $\mathcal{CKG} = \{\mathcal{E}, \mathcal{R}, \mathcal{Q}, \mathcal{EC}, \mathcal{RC}\}$) can be represented as a set of factual quadruples in the form of $(h, r, t, rc) \in \mathcal{Q}$, where $h, t \in \mathcal{E}$, $r \in \mathcal{R}$ and $rc \in \mathcal{RC}$. The notations h and t denote the head and the tail entity of a factual quadruple, r denotes the relations between h and t, and rc denotes . $\mathcal{EC}, \mathcal{RC}$ are the set of entity contexts and relation contexts. Each entity $e \in \mathcal{E}$ and its entity context $ec \in \mathcal{EC}$ form a complete entity representation (e, ec).

To validate whether contextual information can be used to enhance the ability of KG reasoning models, in this paper, we consider two KG reasoning tasks for verification, i.e., KG completion (KGC) and KG question answering (KGQA).

Knowledge Graph Completion Given a query $(h, r, ?)$ or $(?, r, t)$, KGC aims to predict the missing tail or head entity (denoted as "?") that will make the quadruple plausible when the relation context is unknown. Based on the convention of ranking-based evaluation metrics, the aim of a KGC model is to learn a scoring function $f(h, r, t)$ to measure the plausibility of all entities in $\mathcal{E}$ as the missing ones in the query and then rank them in descending order. For performing KGC over a contextual KG, the scoring function $f(h, r, t)$ can be reformulated as $f(h, r, t, hc, rc, tc)$, where $hc \in \mathcal{EC}$, $rc \in \mathcal{RC}$, $tc \in \mathcal{EC}$ denote the contexts of head entity, tail entity and the relation between them, respectively.

Knowledge Graph Question Answering Given a natural question nq and its topic entity $e_{topic} \in \mathcal{E}$, KGQA aims to retrieve related knowledge by generating structured queries or sampling subgraphs from $\mathcal{KG}$ and predict the answer a based on retrieved knowledge, i.e, $a = f(nq, e_{topic}, \mathcal{KG})$. For performing QA over a contextual KG, the prediction function can be reformulated as $f(nq, e_{topic}, \mathcal{CKG})$.

3 Methods

In this section, we introduce KGR[3], a novel KG reasoning paradigm that leverages LLMs to per-

Category	Context Type	Description	Instance
Entity Context	Entity Attribute	Specific properties or characteristics of the entity	Person: height, gender Product: price, color
	Entity Type	Classifications or types to which the entity belongs, providing context within a larger framework or ontology.	Person: actor, artist, scientist, athlete, musician Place: landmark, city, country, state
	Entity Description	Textual descriptions that provide a comprehensive overview of the entity.	Person: A detailed biography or background
	Entity Alias	Alternative names or identifiers for the entity.	Istanbul, alias: Constantinople.
	Entity Reference Link	Links to external resources or webpages that provide additional information about the entity.	Wikipedia pages, official websites, social media profiles, etc.
	Entity Image	Visual representations or photographs of the entity.	Person: photographs or portraits
	Entity Speech	Audio recordings or sounds associated with the entity.	Music audio, audio introductions, etc.
	Entity Video	Video clips or recordings that feature the entity.	Video interviews, a TED talk, etc.
Relation Context	Temporal Information	The time period during which a relationship is valid or relevant.	(Barack Obama, president of, USA, time: 2009-2017)
	Geographic Location	The physical location associated with a relationship or an event involving entities.	(France national football team, win, 2018 FIFA World Cup, location: Russia)
	Quantitative Data	Specific numerical or quantitative information directly related to the relationship.	(Berkshire Hathaway, shareholder of, Apple Inc, Quantity: 790 million shares)
	Provenance information	References to the origin or source of the relationship data.	Documents, news, articles, images, datasets, etc.
	Confidence Level	Indicators of the reliability or confidence in the relationship data.	The accuracy of the relation extraction model
	Event-specific Detail	Information about specific events that define or influence the relationship between entities.	(Argentina national football team, win, France national football team, event: 2022 FIFA World Cup)
	Supplementary Information	Information that provides background or additional context to the relationship, explaining its significance or implications.	News topics, comments, read counts, share counts, like counts, etc.

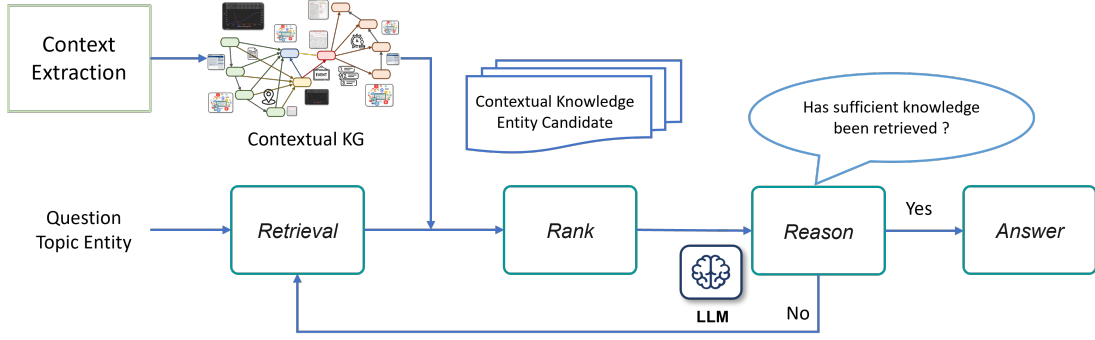Table 1: Examples of different types of entity and relation contexts.

Figure 3: The pipeline of the KGR^3 paradigm.

form KG reasoning tasks based on tructured and contextual semantics. We aim to utilize the complementary relationship between both semantics to improve the reliability and explainability of the reasoning process.

As shown in For triple-based KGs, we begin by augmenting the KG with necessary contextual information extracted from relevant databases, a step that can be omitted if the KG is already a contextual KG. The KGR^3 paradigm consists of three main steps: The **Retrieval** step is to retrieve candidate entities and related contexts from the contextual KG based on the given question; the **Ranking** step involves ranking candidate entities based on the contexts and the given question; the **Reasoning** step is to exploit the LLM to determine whether sufficient information is retrieved. If sufficient information is available, the answer will be generated. If not, the whole processes iterates by retrieving new information based on the top-ranked candidate entities. We give a detailed description of the proposed context-aware paradigms for the KGC and the KBQA tasks

3.1 Context Extraction

Currently, commonly used KG datasets, such as FB15k237, YAGO3-10, and Wikidata5M, are encyclopedic KGs that encapsulate general knowledge about the real world. These KGs are typically developed by domain experts by applying named entity recognition and relation extraction techniques on Wikipedia. However, during this construction process, the rich contexts surrounding the entities are often omitted. Recent studies (Wang et al., 2021, 2022b) have proposed to incorporate entity labels and descriptions as supplementary information for KGs. Nevertheless, the labels and descriptions are insufficient to replace the specific contexts associated with KG triples, thereby limiting their

effectiveness in addressing diverse KG reasoning problems.

To incorporate related contexts into KGs, we consider using Wikidata and Wikipedia as our primary contextual corpus in this work. Due to the extensive coverage and up-to-date information of Wikidata, some KGs like Freebase and YAGO provides official mapping files which can map their entities to Wikidata QIDs. For entities in other KGs, we can use entity search engines provided by Wikidata to find the Wikidata entities which are most likely to be identical to searched entities. Furthermore, Wikidata provides links to the associated Wikipedia pages of its entities. Thus, we can provide contextual information from Wikidata and Wikipedia to different KGs.

3.1.1 Entity Context Extraction

We start to complement the context of a KG with its entities. Specifically, we map the entities from Freebase, YAGO or other KBs to Wikidata QIDs by using official mapping files or using entity search engine provided by Wikidata. For each entity $e_i \in \mathcal{E}$, we collect the textual entity label, the short description, and aliases from Wikidata URIs as its entity context $ec_i \in \mathcal{EC}$. Moreover, the associated Wikipedia pages of Wikidata entities offer vital contextual support for the entities in the KGs. For each entity e_i, we integrate the Wikipedia pages as a part of entity contexts ec_i.

3.1.2 Relation Context Extraction

For each triple, we aggregate the Wikipedia pages of its head and tail entities into a single document. Subsequently, we utilize Sentence-BERT (Reimers and Gurevych, 2019) to identify top-γ supporting sentences that best reflect the semantics of the triple from this document. These sentences not only restore the contexts omitted during the KG construction but also provide optimal support for language
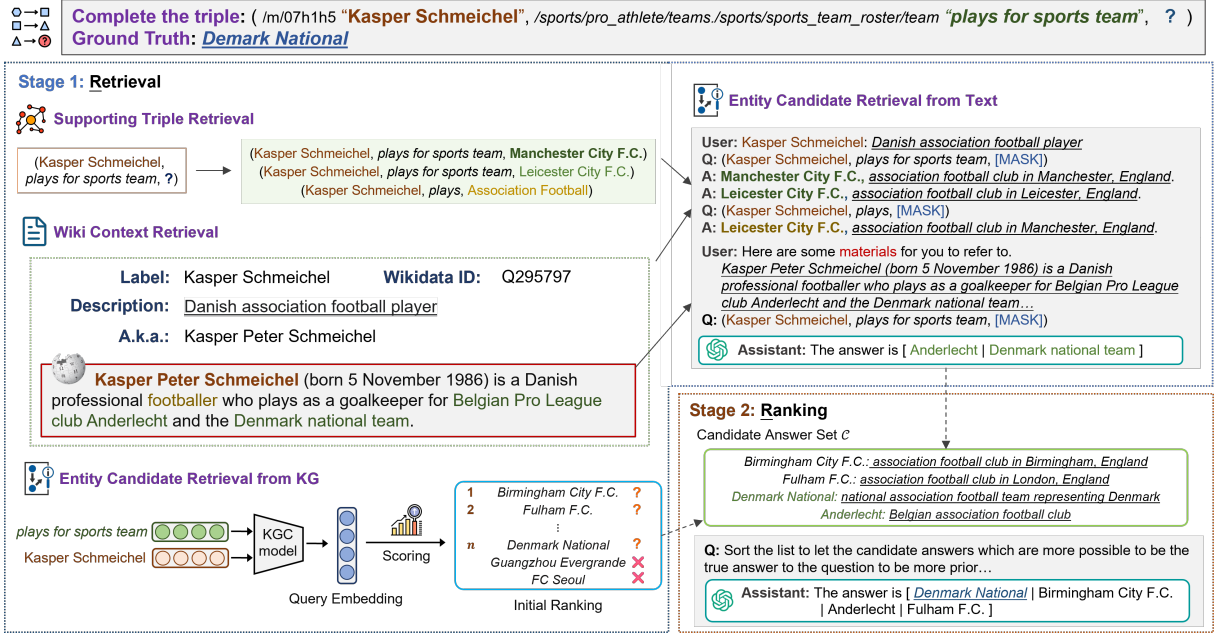
Figure 4: Knowledge Graph Completion.

models in understanding the structured KG triples. Thus, we can regard these supporting sentences as a kind of provenance information or supplementary information and treat them as relation contexts of triples. On the other word, for each triple (h, r, t), we use its supporting sentences extracted from Wikipedia as its relation context $rc \in \mathcal{RC}$ and reshape this triple to a context-aware quadruple (h, r, t, rc).

3.2 Knowledge Graph Completion

In this section, we demonstrate a new context-enriched KGC method based on our proposed KGR^3 paradigm. Since KGC can be considered as an entity ranking task for single-hop reasoning questions, it is not necessary to perform iterative reasoning processes. Thus, the **reasoning** step is omitted for this task.

3.2.1 Step 1: Retrieval

The retrieval module focus on gathering structural and semantic knowledge that may contribute to the completion of certain incomplete triple.

Supporting Triple Retrieval. In KGs, the attributes of an entity are represented in structural triples. Different entities connected by the same relation often share common salient properties. These internal knowledge inherent in the graph structure provide the most direct support to the validity of a triple. Given an incomplete query triple in the form of $(h, r, ?)$ or $(?, r, t)$, we aim

to retrieve k supporting triples that are the most semantically similar to the incomplete query triple. Intuitively, we prioritize triples with the same entity and relation from the training set. If the number of available triples is less than k, we broaden our choices to triples with the same relation, and with entities similar to the known one in the query triple.

Textual Context Retrieval. We note that there is a significant semantic gap between structural triples and natural language. For example, in Figure 4, entity "Kasper Schmeichel" is originally represented by entity id "*/m/07h1h5*" while relation "plays for sports teams" is originally represented as "*/sports/pro_athlete/teams./sports/sports_team _roster/team*". Such a structured format is difficult for LLMs to process. To fully leverage the semantic understanding capabilities of LLMs, we extract relevant contexts related to entities in the query triple and supporting triples from Wikidata knowledge base (Tharani, 2021).

In mainstream KGs, entities are represented in numerical or textual IDs. Each entity ID acts as an index to the data frame in its corresponding KB. Apart from triples, the data-frame of an entity contains significant contextual information such as entity label. To enhance data consistency across different KBs, identical entities across different KBs are aligned with the "owl:sameAs" property. Given its extensive coverage and up-to-date information, Wikidata is employed as our primary contextual

corpus. Specifically, for each entity, we map the entity ID to Wikidata QID with the "owl:sameAs" property. [*] We then collect the textual entity label, the short description, and aliases from Wikidata URIs. Furthermore, Wikidata provides links to the associated Wikipedia pages of its entities. Considering the length of the document, we collect the first paragraph of these Wikipedia pages, which offer complementary semantic support for the completion of query triples.

Candidate Answer Retrieval from KG. The widely adopted ranking-based evaluation for KGC task requires the model to score the plausibility of each entity in the KG as a potential replacement for the missing entity in the query triple. However, given the vast number of entities in the KG, employing LLMs to score and rank each entity is computationally expensive and impractical. Inspired by (Lovelace et al., 2021; Wei et al., 2023; Li et al., 2024), we employ an embedding-based KGC model to initialize the scoring and ranking of entities within the KG. Here, we denote the ranked entity list as $\mathcal{A}_{\text{KGE}} = [e_1^{(k)}, e_2^{(k)}, ..., e_n^{(k)}, ..., e_{|\mathcal{E}|}^{(k)}]$, where the scoring function f_r ensures a descendent ranking order. Formally, we have $f_r(h, e_i^{(k)}) < f_r(h, e_j^{(k)})$ if and only if $i > j$.

Candidate Answer Retrieval from Text. Apart from supporting triples, the Wikipedia page of the known entity also entails rich semantic knowledge. Different from the short Wikidata description, the first Wikipedia paragraph provides a brief introduction to the entity. We anticipate that LLMs can harness their information extraction and comprehension capabilities by utilizing comprehensive contextual information about the known entity, thereby generating potential answers. Specifically, we pass the Wikipedia paragraph of the known entity and the natural language question translated from the query triple to the LLM. Based on the task-specific prompts, the LLM will output a list of answers in its response. However, it should be noted that generative LLMs do not guarantee that output answers will conform to entities in the KG. Therefore, we post-process the LLM output by replacing entity aliases with entity labels and filtering out invalid and unreliable answers that do not appear within the top-δ positions of $\mathcal{A}_{\text{emb}}$. Finally, we obtain a list

of m answers $\mathcal{A}_{\text{LLM}} = [e_1^{(l)}, e_2^{(l)}, ..., e_m^{(l)}]$, where $e_1^{(l)}, e_2^{(l)}, ..., e_m^{(l)} \in \mathcal{E}$, each of which is simultaneously supported by the LLM and the embedding model.

3.2.2 Step 2: Ranking

Motivated by the complementary nature of semantic and structural knowledge, we aim to exploit the candidate answer list generated by the LLM and the KGE model to compose our rankings. To guide the LLM in utilizing entity descriptions for ranking candidate answers to query triples, we introduce supervised fine-tuning (SFT) with LoRA adaptation (Chao et al., 2024). The training objective of SFT is to restore the original plausibility-based ranking for a list of shuffled candidate answers. Specifically, we construct training samples by corrupting the tail (or head) entity of each triple in the validation set. For each corrupted triple, we utilize an embedding-based model to initialize a ranked entity list and collect the top-n entities as candidate answers. Then, we add the ground truth entity to the front of the candidate answer list, and shuffle the list randomly. After that, we translate the masked triple to a question, and retrieve the entity label and the short Wikidata description for each candidate answer. Finally, we provide these questions along with their candidate answers and descriptions to the LLM for training. The LLM will learn to rank the candidate answers based on their contextual relevance and plausibility by considering the semantics of the question and entity descriptions.

During the inference stage, we construct a candidate answer set $\mathcal{C}$ with top-n entities from $\mathcal{A}_{\text{KGE}}$ and all entities in $\mathcal{A}_{\text{LLM}}$. Formally, we have $\mathcal{C} = \mathcal{A}_{\text{KGE}}[0 : n] \cup \mathcal{A}_{\text{LLM}}$. Then we employ the fine-tuned LLM to re-rank entities in $\mathcal{C}$ with their descriptions and the LLM's intrinsic knowledge. Subsequently, the LLM will output a re-ordered answer list $\mathcal{A}_{\text{RR}} = [e_1^{(o)}, e_2^{(o)}, ..., e_{|\mathcal{C}|}^{(o)}]$. Finally, we remove all entities in $\mathcal{C}$ from the original entity list $\mathcal{A}_{\text{KGE}}$, and compose the final ranking of all entities by attaching $\{\mathcal{A}_{\text{KGE}} \setminus \mathcal{C}\}$ to the end of $\mathcal{A}_{\text{RR}}$.

3.3 Knowledge Base Question Answering

In this section, we introduce an in-context learning paradigm for the KBQA task (see Figure 5). This paradigm focuses on the integration of contextual information, which plays a pivotal role in identifying plausible reasoning paths and facilitating the derivation of final answers.

---

[*]Since Google Freebase is deprecated and migrated to Wikidata, we map the entity IDs in the FB15k237 dataset to corresponding Wikidata QIDs with official data dumps.
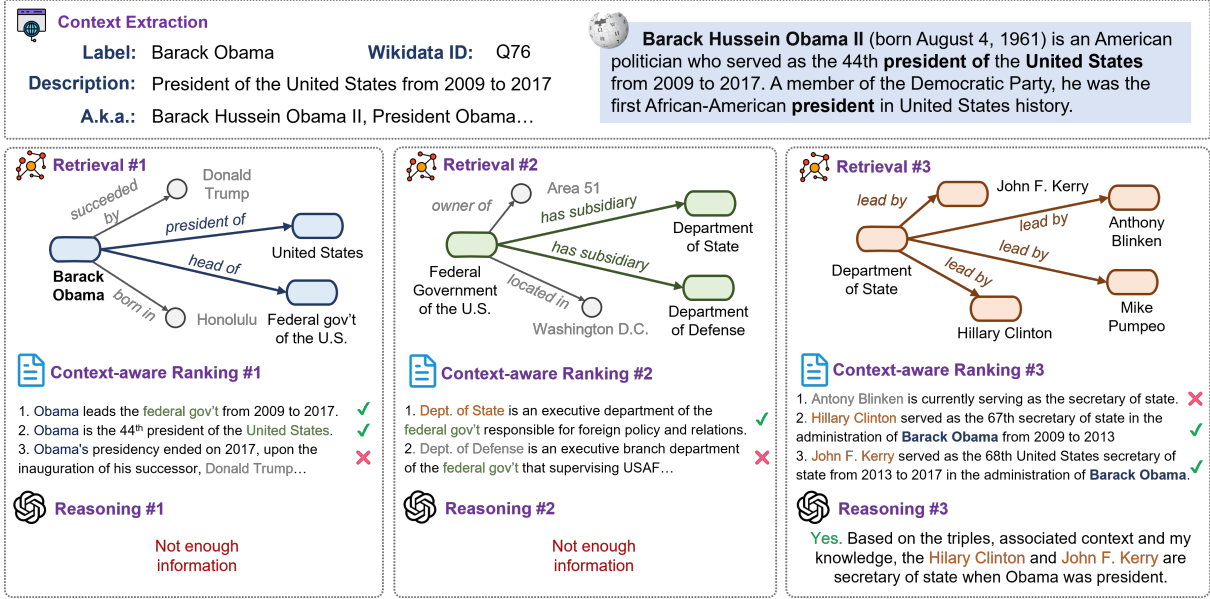
Figure 5: Knowledge Base Question Answering.

Given a question q, we first identifies a set of k topic entities $E^{(0)} = \{e_i^{(0)}\}_{i=1}^k$ with an LLM. Starting from these topic entities, we iteratively explore plausible reasoning paths until the LLM determines that it can answer the question based on the support of triples along the paths and their associated contexts. Therefore, during the inference process, we maintain and update a set of reasoning paths $P = \{p_1, p_2, ..., p_M\}$ alongside a list of relation context sentences $C = \{rc_1, rc_2, ..., rc_N\}$. Here, M represents the width of the beam search, while N denotes the number of relation context sentences. Each iteration of the process consists of three steps: 1) knowledge exploration, 2) reasoning path pruning, and 3) context-aware reasoning.

At the beginning of the D-th iteration, each reasoning path consists of $D-1$ triples, i.e., $p_i = \{(h_n^{(d)}, r_n^{(d)}, t_n^{(d)})\}_{d=1}^{D-1}$, where $h_n^{(1)}$ is a topic entity from $E^{(0)}$, $t_n^{(d)} = h_n^{(d+1)}$ ensures the tail entity of one triple becomes the head entity of the next. [†]

3.3.1 Step 1: Context-aware Triple Retrieval

In the initial step, we aim to retrieve candidate triples that can extend the reasoning paths. Specifically, for each reasoning path $p_m \in P$, we collect the tail entity $e_m^{(D-1)}$ from the last triple and identify the set of relations $R_m^{(D)}$ linked to the

---

[†]WLOG, We only look for paths with forward relations. For each triple (h, r, t), we introduce a reversed relation r^{-1} and the reversed triple (t, r^{-1}, h) into the KG.

entity. We then construct queries in the form of $(e_m^{(D-1)}, r_m^{(D)}, ?)$ using each of the relations . Given that an entity can be linked to multiple relations, this process potentially increases the number of reasoning paths. To reduce the computational complexity, we exploit the LLM to select top-M queries based on their relevance to the question. Subsequently, we proceed to complete the query triples by retrieving suitable neighboring entities from the KG, each of which derives a candidate triple that can potentially lead to answering the question.

3.3.2 Step 2: Candidate Entity Ranking

In the second step, we focus on identifying those triples that are most likely to contribute to a correct answer. First, we augment each candidate triple with γ relation context sentences that are best aligned with its contextual semantics as described in Section 3.1. With relation contexts, we then exploit the LLM to select out top-M triples from the candidate triples derived from each query $(e_m^{(D-1)}, r_m^{(D)}, ?)$. This helps us to prune out irrelevant and noisy neighboring entities that could mislead the LLM into producing incorrect answers. Due to the length limit of LLM inputs, it is still impractical to leverage the remaining $M \times M$ triples in knowledge reasoning. Therefore, we further refine our selection from the remaining triples to top-M triples with the highest contextual relevance

between the relation contexts and the question [‡]. Finally, we attach the M triples to the end of each corresponding reasoning path and append their relation contexts into the context list C. The context list C are then updated by ranking their relevance to the given question and only top-N relation context sentences are remained at the end of this step.

3.3.3 Step 3: Context-aware Reasonin

Upon obtaining the new top-M reasoning paths P and updating relation context list C, this extra knowledge retrieved from the contextual KG are integrated into the origin question as a part of the prompt. The prompt is input to the LLM and the LLM perform the reasoning step to determine whether the sufficient information has been retrieved from the contextual KG. If yes, the LLM generates the answer based on the retrieved knowledge and its inherent knowledge. Otherwise, the whole process will iterate by starting the first step with new reasoning paths P and relation context set C.

4 Experiments on KG Completion

In this section, we assess the effectiveness of KGR^3 in the KGC task. Our investigation is guided by the three following research questions:

- **RQ1:** Whether KGR^3 works for varied embedding methods?

- **RQ2:** Whether different types of entity contexts contribute to enhancing KG reasoning?

- **RQ3:** Can LLM effectively leverage entity contexts for the KGC task with or without SFT?

- **RQ4:** Can KGR^3 improve the inference performance for predicting long-tail entities?

4.1 Datasets

We evaluate our proposed framework on two widely-used datasets FB15k237 (Toutanova et al., 2015) and YAGO3-10 (Rebele et al., 2016). FB15k237 is derived from Freebase (Bollacker et al., 2008), an encyclopedic knowledge base containing general knowledge about topics such as celebrities, organizations, movies, and sports. YAGO3-10 is a subset of YAGO3 (Rebele et al., 2016), a knowledge base built upon Wikipedia, WordNet (Miller, 1995), and GeoNames (Bond

and Bond, 2019). To prevent potential data leakage, FB15k237 excludes reversible relations from the backend KB. Detailed statistics of the two datasets are shown in Table 2.

Dataset	FB15k237	YAGO3-10
#Entities	14,541	123,182
#Relations	237	37
#Train	272,115	1,079,040
#Valid	17,535	5,000
#Test	20,466	5,000

Table 2: Statistics of Datasets

4.2 Baselines

In this section, we evaluate the efficacy of our proposed KGR^3 framework by integrating it with three widely utilized embedding-based KGC models: RotatE (Sun et al., 2019), ComplEx (Trouillon et al., 2016), and GIE (Cao et al., 2022). These models not only serve as baseline methods but are also foundational for candidate answer retrieval. Instead of surpassing all baseline methods, our main objective is to evaluate the effectiveness of our context-enriched KGR^3 framework when applying to different embedding models. Hence, we deliberately include a limited selection of baseline models.

4.3 Implementation Details

We conduct all of our experiments on a Linux server with two Intel Xeon Platinum 8358 processors and eight A100-SXM4-40GB GPUs. We choose the framework provided by the GIE (Cao et al., 2022) project for training the base embedding models, strictly following the parameter settings provided. During the reasoning stage, we utilize OpenAI's gpt-3.5-turbo-0125 checkpoint [§]. The Re-ranking stage employs Meta-Llama-3-8B-Instruct with BF16 precision as the backbone model [¶]. The SFT task is implemented based on the LLaMA-Factory (Zheng et al., 2024) framework and applies LoRA technique (Hu et al., 2021), with a rank setting to 16 and an alpha setting to 32. Additionally, AdamW (Loshchilov and Hutter, 2017) is used as the optimizer, the batch size is set to 2 per device, the gradient accumulation steps is set to 4, and the learning rate is 1.0e-4. The sampling ratio of the validation set is 5%, and the best checkpoint is selected based on evaluation loss.

[‡]We utilize the bge-large-en-v1.5 model to measure the semantic similarity of the question and each supporting sentence.

[§]https://platform.openai.com/docs/models

[¶]https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

Model	FB15K-237				YAGO3-10			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
ComplEx	0.247	0.158	0.275	0.428	0.360	0.260	0.400	0.550
ComplEx + KGR^2	0.315	0.248	0.343	0.428	0.402	0.336	0.430	0.537
ComplEx + KGR^3	**0.333**	**0.263**	**0.365**	**0.460**	**0.408**	**0.340**	**0.441**	**0.559**
Improvements	34.82%	66.46%	32.73%	7.48%	13.33%	30.77%	10.25%	1.64%
RotatE	0.338	0.241	0.375	0.533	0.495	0.402	0.550	0.670
RotatE + KGR^2	0.370	0.283	0.404	0.542	0.508	0.422	0.553	0.662
RotatE + KGR^3	**0.382**	**0.293**	**0.417**	**0.559**	**0.521**	**0.443**	**0.572**	**0.678**
Improvements	13.02%	21.58%	11.20%	4.88%	5.25%	10.20%	4.00%	1.19%
GIE	0.362	0.271	0.401	0.552	0.579	0.505	0.618	**0.709**
GIE + KGR^2	0.378	0.288	0.412	0.557	**0.599**	**0.522**	**0.633**	0.702
GIE + KGR^3	**0.391**	**0.301**	**0.426**	**0.573**	0.597	0.518	0.625	0.698
Improvements	8.01%	11.07%	6.23%	3.80%	3.45%	3.37%	2.43%	-0.99%
Avg. Improvements	18.62%	33.04%	16.72%	5.39%	7.34%	14.78%	5.56%	0.61%

Table 3: Experiment results of the KGC task on FB15k-237 and YAGO3-10 datasets. The best results are in **bold**.

4.4 Evaluation

For each query triple in the form of $(h, r, ?)$ or $(?, r, t)$, the KGC model outputs a ranked list of all entities in the KG. For a fair comparison, we adopt the "filtered" setting introduced in (Bordes et al., 2013). Except for the ground truth entity, we remove all other valid entities that conform to an existing triple in training, validation, or test set from the ranked list in advance. Based on the position of the ground truth entity, We compute Hits@1, Hits@3, Hits@10 and mean reciprocal rank (MRR), where higher results indicate better performance.

4.5 Main Results

Table 3 summarizes the performance of the KGR^3 framework on three different base embedding methods. The experiment results show that KGR^3 and its simplified variants KGR^2 without "*reasoning*" module significantly and consistently enhances each embedding method among all metrics. On average, our KGR^3 framework improves the Hits@1 by 33.04% and 14.78% on FB15k-237 and YAGO3-10 datasets. These results demonstrate the effectiveness and superiority of integrating LLMs and entity contexts with embedding-based KGC models, which address our **RQ1**.

Notably, the improvement in Hits@1 is more substantial than that in Hits@3 and Hits@10. This indicates that the KGR^3 framework is particularly effective at identifying the most accurate answers. Since our framework primarily focuses on re-ordering top-n (or top-δ if we consider reasoning outputs) entities from the initial ranked entity list, the upper bound of Hit@1, Hit@3, and Hit@10 are implicitly constrained by the Hits@n or Hit@δ performance of the base embedding model. Given that Hits@1 is typically further from this upper bound, the potential for improvement will be greater. Additionally, by leveraging semantic knowledge from entity contexts, the LLM gains a more comprehensive understanding of the entities, thereby enabling more precise inferences, particularly for top-ranked candidate answers.

Furthermore, the performance gains are more pronounced for simpler embedding models such as ComplEx (Trouillon et al., 2016). Simple embedding models cannot fully capture the structural information in the KG, leading to the introduction of noisy entities in the candidate answer list. With entity descriptions, the LLM can utilize its semantic understanding capabilities to identify and deprioritize candidate answers that do not match the semantics of the query triple. Hence, KGR^3 can enhance the robustness of these embedding models.

In addition, a comparison between KGR^2 and KGR^3 reveals that the inclusion of the "*reasoning*" provides a notable boost. In certain scenarios, the KG may lack sufficient structural information to derive plausible answers. Nevertheless, long Wikipedia paragraphs can effectively augment specific entities with extra semantic knowledge, which allows the LLM to generate additional candidate answers with its semantic reasoning capability. This surpasses the inherent limitations of KGs, leading to substantial performance improvements. An case study showing the effectiveness of the **Reasoning** and **Re-ranking** processes is demontrated in Appendix A.1-A.3

11

4.6 Ablation Studies

4.6.1 Effectiveness of Entity Contexts

To address **RQ2**, we assess the contribution of different types of contexts in the reasoning and re-ranking modules of KGR^3, and conduct ablation studies on FB15k-237 dataset. In the "KGR^3 w/o context in Reasoning" variant, we remove the short descriptions used to explain the entities and replace the Wikipedia paragraph of the known entity with an entity label. Under such circumstances, the LLM cannot fully demonstrate its strong semantic understanding capability, resulting in lower performance.

In the "KGR^3 w/o context in Re-ranking" variant, we simply remove the entity descriptions for each candidate answer, which results in a noticeable performance decline. This decline reveals that LLMs may lack a fundamental understanding of certain entities within the KG. Consequently, without sufficient semantic information, the LLM cannot rank candidate answers effectively.

If we remove all contextual information from the KGR^3 framework, performance deteriorates even further. This indicates that every type of context is meaningful and irreplaceable, playing a crucial role at each stage of the process. Without entity contexts, the LLM only relies on its inherent knowledge, hence leading to suboptimal inference results.

With proper base embedding model, KGR^3 surpasses the state-of-the-art embedding-based model CompoundE (Ge et al., 2023) and the text-based model SimKGC (Wang et al., 2022a). This demonstrates that entity context can compensate for the limitation of embedding methods in modelling the graph structure. Furthermore, discrepancies between SimKGC and KGR^3 underscores the limitations of existing text-based methods. On the one hand, PLM-driven models exhibit insufficient semantic understanding, and the gap between lightweighted PLM and LLM cannot be easily alleviated by fine-tuning. On the other hand, these methods underutilize the semantic and structural information within KG. When being applied to complete a specific triple, they often consider the triple in isolation, neglecting the local neighborhood of the known entity and other similar triples.

4.6.2 Effectiveness of SFT

In response to **RQ3**, we conduct extra experiments on KGR^3 with different LLMs. From Table 5, we observe that if we remove SFT step from the re-

Settings	MRR	Hits@1	Hits@3	Hits@10
ComplEx + KGR^3	**0.333**	**0.263**	**0.365**	**0.460**
- w/o context in **Reasoning**	0.330	0.260	0.361	0.454
- w/o context in **Re-ranking**	0.319	0.245	0.351	0.453
- w/o all contexts	0.305	0.235	0.336	0.428
RotatE + KGR^3	**0.382**	**0.293**	**0.417**	**0.559**
- w/o context in **Reasoning**	0.375	0.285	0.411	0.555
- w/o context in **Reranking**	0.361	0.264	0.398	0.559
- w/o all contexts	0.360	0.262	0.398	0.561
GIE + KGR^3	**0.391**	**0.301**	**0.426**	**0.573**
- w/o. context in **Reasoning**	0.384	0.290	0.422	0.574
- w/o. context in **Re-ranking**	0.366	0.267	0.403	0.572
- w/o. all contexts	0.363	0.267	0.400	0.556
CompoundE	0.357	0.264	0.393	0.545
SimKGC	0.336	0.249	0.362	0.511

Table 4: Ablation Experiments on FB15k-237 dataset with different combinations of contexts.

Settings	MRR	Hits@1	Hits@3	Hits@10
ComplEx + KGR^3	**0.329**	**0.256**	**0.363**	**0.456**
- w/ non-SFT Llama3	0.288	0.206	0.323	0.450
- w/ ChatGPT	0.299	0.224	0.330	0.453
RotatE + KGR^3	**0.380**	**0.287**	**0.417**	**0.565**
- w/ non-SFT Llama3	0.321	0.215	0.356	0.556
- w/ ChatGPT	0.348	0.248	0.387	0.559
GIE + KGR^3	**0.383**	**0.291**	**0.418**	**0.576**
- w/ non-SFT Llama3	0.324	0.213	0.364	0.564
- w/ ChatGPT	0.354	0.253	0.391	0.570
KICGPT w/ limited demos	0.274	0.183	0.280	0.496

Table 5: The performance of KGR^3 without SFT on the first 2,000 examples of FB15k-237 dataset.

ranking module, the performance significantly decreases, even potentially falling below base embedding models. Despite with certain semantic understanding capabilities, vanilla LLMs cannot perform well in ranking tasks. We can further conclude that the ability to perform ranking based on entity context is acquired during the fine-tuning process. Compared to Llama, ChatGPT achieves a better performance with its stronger instruction following capability. Nevertheless, ChatGPT still lags far behind the finetuned Llama, showcasing the necessity of SFT.

Moreover, we compare KGR^3 with state-of-the-art LLM-based KGC baseline KICGPT [‖]. It should be noted that KICGPT processes all triples in the KG with the same entity or relation as the incomplete triple, which consumes far more $(20\times)$ tokens than our KGR^3 framework. For a fair comparison, we re-evaluate KICGPT with k supporting triples. From the experimental results we observe

---

[‖]We only modify the parameters `demo_per_step` to 2, `max_demo_step` to 2 and `candidate_num` to 10 in (Wei et al., 2023), to ensure the consistency with the settings in this work. Since there is no metric evaluation provided, we evaluated the natural language results generated within our framework.
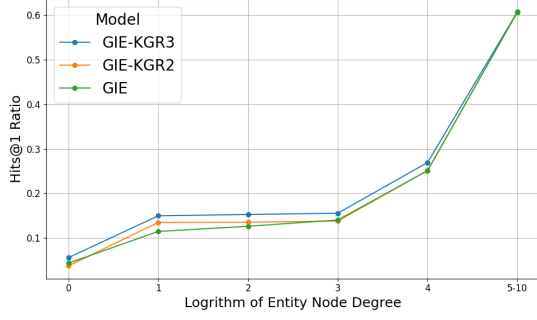
Figure 6: Average Hit@1 performance of GIE, GIE-KGR^2 and GIE-KGR^3 grouped by the logarithm of entity node degree on FB15k-237 dataset.

that KICGPT left significantly behind all variants of KGR^3. The remarkable performance gap can also be explained by the introduction of SFT since KICGPT employs ChatGPT as its backbone.

4.6.3 Effect on Handling Long-tail Entities

In response to **RQ4**, we follow (Wang et al., 2022c; Wei et al., 2023) and group triples from FB15k-237 test set into 5 classes with the logarithm of the node degree of their known entities. We average the Hit@1 performance of each group of triples with KGR^3, KGR^2 (w/o reasoning module), and their base embedding model GIE (Cao et al., 2022) (see Figure 6). From Figure 6 we observe that KGR^3 consistently outperforms its variant KGR^2 and GIE in all groups, especially for the first two groups where entities have fewer neighbors. This empirically shows that the proposed framework can effectively alleviate the long-tail problem. In addition, the performance gap between KGR^2 and GIE is less pronounced, which reaffirms the importance of the reasoning part, where the LLM generates possible answers based on the Wikipedia introduction of entities.

5 Experiments on KGQA

5.1 Datasets and Evaluation Metric

We note that a lot of commonly-used KGQA benchmarks like CWQ (Talmor and Berant, 2018) and WebQSP (Yih et al., 2016) are constructed from Freebase (Bollacker et al., 2008) which has been defunct since 2015. Some of the knowledge in Freebase is outdated or contradicts information in Wikipedia (Xu et al., 2023). Clearly, compared to Freebase, the knowledge in Wikipedia has higher coverage and accuracy, and in this work, Wikipedia serves as the main source of contextual information. Our assumption is that the contextual information

can support or complement the triple-based knowledge in the KG, rather than contradict it. Therefore, we consider KGQA datasets based on Wikidata where the triple-based knowledge is better aligned with the contextual information from Wikipedia, rather than KGQA datasets constructed from Freebase.

In this work, QALD10-en (Usbeck et al., 2023) and WikiWebQuestion (WWQ) (Xu et al., 2023) are used as KGQA datasets for evaluation. QALD10-en is a new, complex, Wikidata-based KGQA benchmarking dataset as the 10th part of the Question Answering over Linked Data (QALD) benchmark series. WWQ is constructed by migrating the popular WebQSP (Yih et al., 2016) benchmark from Freebase to Wikidata, with updated SPARQL and up-to-date answers from the much larger Wikidata.

For all datasets, exact match accuracy (EM) is used as our evaluation metric following previous works (Li et al., 2023; Sun et al., 2024).

5.2 Baseline

We compare with standard prompting (IO prompt) (Brown et al., 2020b), Chain-of-Thought prompting (CoT prompt) (Wei et al., 2022), and Self-Consistency (Wang et al., 2023) with 6 in-context exemplars and "step-by-step" reasoning chains. Moreover, for each dataset, we pick previous state-of-the-art (SOTA) works for comparison. We notice that fine-tuning methods trained specifically on evaluated datasets usually have an advantage by nature over methods based on prompting without training, but sacrificing the flexibility and generalization on other data. Therefore, we compare with previous SOTA among all prompting-based methods and previous SOTA among all fine-tuned (FT) methods respectively. With regard to previous prompting-based methods, we select their results achieved with GPT-3.5 for a a fair play.

5.3 Implementation

We use ChatGPT (GPT-3.5-turbo) as the backbone LLM for KGR^3 by calling OpenAI API. The maximum token length for the generation is set to 256. In all experiments, we set both width M and depth D_{max} to 3 for beam search. We use 5 shots in KGR^3-reasoning prompts for all the datasets.

5.4 Experimental Results

Since KGR^3 uses external KGs and contextual information to enhance LLM, we first compare it

Method	QALD10-en	WWQ
Without external knowledge		
IO prompt w/ChatGPT	42.0	57.7
SC w/ChatGPT	42.9	-
SC w/ChatGPT	45.3	-
With external knolwedge		
Prior FT SOTA	45.4^{α}	65.5^{β}
Prior Prompting SOTA	50.2^{θ}	72.6^{θ}
Ours		
KGR^3	**54.7**	**78.8**
KGR^3 w.o./Context	38.1	67.3
Gain	(+43.6)	(+17.1)

Table 6: Exact match accuracy of KGR^3 using ChatGPT as the backbone models on QALD10-en and WWQ. The prior FT (Fine-tuned) and prompting SOTA include the best-known results: α: (Santana et al., 2022); β: (Xu et al., 2023); θ: Sun et al. (2024)

.

with those methods leveraging external knowledge as well. As we can see in Table 6, even if KGR^3 is a training-free prompting-based method and has natural disadvantage in comparison with those fine-tuning methods trained with data for evaluation, KGR^3 still achieves new SOTA performance in both datasets. If comparing with other promoting-based methods with ChatGPT as backbone models (especially ToG), KGR^3 can win the competition on all datasets.

It is noteworthy that other prompting-based methods rely solely on triple knowledge from KGs, whereas KGR^3 allows the LLM to leverage additional contextual information for more precise reasoning on KGs. This is likely the primary reason why KGR^3 outperforms other prompting-based methods. To verify this, we evaluated a variant of KGR^3 that excludes contextual information for comparison. As shown in Table 6, incorporating contextual information results in a relative increase of 43.6% and 17.1% in Exact Match (EM) on QALD10-en and WWQ, respectively. These experimental results support our hypothesis that KGQA methods can significantly benefit from the integration of contextual information.

6   Conclusion

This work points out several critical shortcomings of triple-based KGs, including their inability to represent diverse knowledge flexibly and perform complex knowledge reasoning accurately, due to the lack of contextual information. By highlighting

these limitations, we underscore the necessity of moving beyond triple-based representation for KGs and introduce the concept of contextual KGs. Contextual KGs integrate rich contextual data, such as temporal, geographic, and provenance information, thus providing a more comprehensive and accurate representation of knowledge. This enhanced representation supports more effective reasoning by leveraging the added layers of contextual information.

To verify the effectiveness of incorporating contextual KGs on KG representation and reasoning, we present KGR^3, a novel reasoning paradigm for KGs that integrates LLMs (LLMs) with contextual KGs to address the limitations of traditional triple-based KG reasoning methods. Through extensive experiments on KG completion and KG question answering tasks, we demonstrated that incorporating contextual information significantly improves the performance of existing models. Our results underscore the importance of context in capturing the complexity and richness of real-world knowledge, enabling more nuanced and accurate inferences.

In conclusion, the introduction of contextual KGs represents a significant step forward in the evolution of KGs, offering a more sophisticated and comprehensive approach to knowledge representation and reasoning. This work opens new avenues for future research and applications, highlighting the potential of contextual KGs and LLMs in advancing the field of artificial intelligence.

References

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, page 1247–1250, New York, NY, USA. Association for Computing Machinery.

Francis Bond and Arthur Bond. 2019. GeoNames Wordnet (geown): extracting wordnets from GeoNames. In *Proceedings of the 10th Global Wordnet Conference*, pages 387–393, Wroclaw, Poland. Global Wordnet Association.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26*, volume 26. Curran Associates, Inc.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind

Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020a. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

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. 2020b. 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*.

Zongsheng Cao, Qianqian Xu, Zhiyong Yang, Xiaochun Cao, and Qingming Huang. 2022. Geometry interaction knowledge graph embeddings. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(5):5521–5529.

Wenshuo Chao, Zhi Zheng, Hengshu Zhu, and Hao Liu. 2024. Make large language model a better ranker. *Preprint*, arXiv:2403.19181.

Xin Luna Dong. 2023. Generations of knowledge graphs: The crazy ideas and the business impact. *arXiv preprint arXiv:2308.14217*.

Xiou Ge, Yun Cheng Wang, Bin Wang, and C.-C. Jay Kuo. 2023. Compounding geometric operations for knowledge graph completion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6947–6965, Toronto, Canada. Association for Computational Linguistics.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, and Christian Bizer. 2014. Dbpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, 6.

Muzhi Li, Minda Hu, Irwin King, and Ho fung Leung. 2024. The integration of semantic and structural knowledge in knowledge graph entity typing. *Preprint*, arXiv:2404.08313.

Xingxuan Li, Ruochen Zhao, Yew Ken Chia, Bosheng Ding, Shafiq Joty, Soujanya Poria, and Lidong Bing. 2023. Chain-of-knowledge: Grounding large language models via dynamic knowledge adapting over heterogeneous sources. In *The Twelfth International Conference on Learning Representations*.

Ruotong Liao, Xu Jia, Yangzhe Li, Yunpu Ma, and Volker Tresp. 2024. Gentkg: Generative forecasting on temporal knowledge graph with large language models. *Preprint*, arXiv:2310.07793.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Justin Lovelace, Denis Newman-Griffis, Shikhar Vashishth, Jill Fain Lehman, and Carolyn 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 (Volume 1: Long Papers)*, pages 1016–1029, Online. Association for Computational Linguistics.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Thomas Pellissier Tanon, Gerhard Weikum, and Fabian Suchanek. 2020. Yago 4: A reason-able knowledge base. In *The Semantic Web: 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31–June 4, 2020, Proceedings 17*, pages 583–596. Springer.

Thomas Rebele, Fabian Suchanek, Johannes Hoffart, Joanna Biega, Erdal Kuzey, and Gerhard Weikum. 2016. Yago: A multilingual knowledge base from wikipedia, wordnet, and geonames. In *The Semantic Web – ISWC 2016: 15th International Semantic Web Conference, Kobe, Japan, October 17–21, 2016, Proceedings, Part II*, page 177–185, Berlin, Heidelberg. Springer-Verlag.

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*. Association for Computational Linguistics.

Manuel Alejandro Borroto Santana, Bernardo Cuteri, Francesco Ricca, and Vito Barbara. 2022. SPARQL-QA enters the QALD challenge. In *Proceedings of the 7th Natural Language Interfaces for the Web of Data (NLIWoD) co-located with the 19th European Semantic Web Conference (ESWC 2022), Hersonissos, Greece, May 29th, 2022*, volume 3196 of *CEUR Workshop Proceedings*, pages 25–31. CEUR-WS.org.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M. Ni, Heung-Yeung Shum, and Jian Guo. 2024. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. *Preprint*, arXiv:2307.07697.

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

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 641–651. Association for Computational Linguistics.

Karim Tharani. 2021. Much more than a mere technology: A systematic review of wikidata in libraries. *The Journal of Academic Librarianship*, 47(2):102326.

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*, pages 1499–1509, Lisbon, Portugal. Association for Computational Linguistics.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Eric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2071–2080, New York, New York, USA. PMLR.

Ricardo Usbeck, Xi Yan, Aleksandr Perevalov, Longquan Jiang, Julius Schulz, Angelie Kraft, Cedric Möller, Junbo Huang, Jan Reineke, Axel-Cyrille Ngonga Ngomo, et al. 2023. Qald-10–the 10th challenge on question answering over linked data. *Semantic Web*, (Preprint):1–15.

Haoran Wang and Kai Shu. 2023. Explainable claim verification via knowledge-grounded reasoning with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6288–6304, Singapore. Association for Computational Linguistics.

Hongwei Wang, Miao Zhao, Xing Xie, Wenjie Li, and Minyi Guo. 2019. Knowledge graph convolutional networks for recommender systems. In *The World Wide Web Conference*, WWW '19, page 3307–3313, New York, NY, USA. Association for Computing Machinery.

Liang Wang, Wei Zhao, Zhuoyu Wei, and Jingming Liu. 2022a. SimKGC: Simple contrastive knowledge graph completion with pre-trained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4281–4294, Dublin, Ireland. Association for Computational Linguistics.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021. KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation. *Transactions of the Association for Computational Linguistics*, 9:176–194.

Xintao Wang, Qianyu He, Jiaqing Liang, and Yanghua Xiao. 2022b. Language models as knowledge embeddings. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 2291–2297. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Xintao Wang, Qianyu He, Jiaqing Liang, and Yanghua Xiao. 2022c. Language models as knowledge embeddings. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 2291–2297. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv Preprint*.

Yanbin Wei, Qiushi Huang, Yu Zhang, and James Kwok. 2023. KICGPT: Large language model with knowledge in context for knowledge graph completion. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8667–8683, Singapore. Association for Computational Linguistics.

Silei Xu, Shicheng Liu, Theo Culhane, Elizaveta Pertseva, Meng-Hsi Wu, Sina Semnani, and Monica Lam. 2023. Fine-tuned llms know more, hallucinate less with few-shot sequence-to-sequence semantic parsing over wikidata. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Wen-tau Yih, Matthew Richardson, Christopher Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*. The Association for Computer Linguistics.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. *arXiv preprint arXiv:2403.13372*.

A   Appendix

A.1   Prompt templates of Retrieval stage

Table 7 shows the prompt templates of the **Retrieval** stage and give an example from FB15k237.

KG Triplet for completion: ([MASK], */location/adjoining_relationship/adjoins*, Champaign)

Task for completion: "The question is to predict the head entity [MASK] from the given ([MASK], *location adjoining_relationship adjoins*, Champaign) by completing the sentence 'Champaign is the adjoins of what location? The answer is '."

Task demonstrations:

Demo 1: "The question is to predict the head entity [MASK] from the given ([MASK], *location adjoining_relationship adjoins*, Washington County) by completing the sentence 'Washington County is the adjoins of what location? The answer is '."
"The answer is Westmoreland County, so the [MASK] is Westmoreland County."

Demo 2: "The question is to predict the head entity [MASK] from the given ([MASK], *location adjoining_relationship adjoins*, Rockland County) by completing the sentence 'Rockland County is the adjoins of what location? The answer is '."
"The answer is Bergen County, so the [MASK] is Bergen County."

Task demonstrations with Contextual Retrieval:

Demo 1: "Washington County: county in Pennsylvania, U.S. The question is to predict the head entity [MASK] from the given ([MASK], *location adjoining_relationship adjoins*, Washington County) by completing the sentence 'Washington County is the adjoins of what location? The answer is '."
"The answer is Westmoreland County, so the [MASK] is Westmoreland County. Westmoreland County: county in Pennsylvania, United States"

Demo 2: "The question is to predict the head entity [MASK] from the given ([MASK], *location adjoining_relationship adjoins*, Rockland County) by completing the sentence 'Rockland County is the adjoins of what location? The answer is '."
"The answer is Bergen County, so the [MASK] is Bergen County. Bergen County: county in New Jersey, United States"

Candidate entities: [Cook County, Champaign, Bloomington, McHenry County, Evanston]

Candidate Answers with Contextual Retrieval:
Cook County: county in Illinois, United States
Champaign County: county in Illinois, United States
Bloomington: city and the county seat of McLean County, Illinois, United States
McHenry County: county in Illinois, United States
Evanston: suburban city in Cook County, Illinois, United States

---

Table 7: Prompt template of retrieval stage.

A.2 Prompt templates of Reasoning stage

Table 8 shows the prompt templates of the **Reasoning** stage and give an example which is the same case as Table 7.

KG Triplet for completion: ([MASK], */location/adjoining_relationship/adjoins*, Champaign)

Task for completion: "The question is to predict the head entity [MASK] from the given ([MASK], *location adjoining_relationship adjoins*, Champaign) by completing the sentence 'Champaign is the adjoins of what location? The answer is '."

Reasoning:

The question is to predict the head entity [MASK] from the given ([MASK], *location adjoining_relationship adjoins*, Champaign) by completing the sentence 'Champaign is the adjoins of what location? The answer is '. Output all some possible answers based on your own knowledge, using the format '[answer1, answer2, ..., answerN]' and please start your response with 'The possible answers:'. Do not output anything except the possible answers.

Context-aware Reasoning:

Here are some materials for you to refer to. Champaign: Champaign is a city in Champaign County, Illinois, United States. The population was 88,302 at the 2020 census. It is the tenth-most populous municipality in Illinois and the fourth most populous city in the state outside the Chicago metropolitan area. It is a principal city of the Champaign–Urbana metropolitan area, which had 236,000 residents in 2020. Champaign shares the main campus of the University of Illinois with its twin city of Urbana, and is also home to Parkland College, which gives the city a large student population during the academic year. Due to the university and a number of technology startup companies, it is often referred to as a hub of the Illinois Silicon Prairie. Champaign houses offices for the Fortune 500 companies Abbott, Archer Daniels Midland (ADM), Caterpillar, John Deere, Dow Chemical Company, IBM, and State Farm. Champaign also serves as the headquarters for several companies, including Jimmy John's.

The question is to predict the head entity [MASK] from the given ([MASK], *location adjoining_relationship adjoins*, Champaign) by completing the sentence 'Champaign is the adjoins of what location? The answer is '. Output all the possible answers you can find in the materials using the format '[answer1, answer2, ..., answerN]' and please start your response with 'The possible answers:'. Do not output anything except the possible answers. If you cannot find any answer, please output some possible answers based on your own knowledge.

Context-aware Reasoning result by LLM:

The possible answers: Urbana, Champaign County, Illinois Silicon Prairie, Parkland College.

Table 8: Prompt Template of context-aware reasoning.

A.3 Prompt templates of Ranking stage

Table 9 shows the prompt templates of the **Retrieval** stage and give an example which is the same case as Table 7 and 8.

Noteworthily, this case also empirically shows the effectiveness of the **Reasoning** and **Re-ranking** processes. The ground truth answer 'Urbana' is not successfully retrieved by the KGC model, GIE. However, the LLM provides new candidates including the ground truth answer 'Urbana', by analyzing the context of the known entity 'Champaign' in the incomplete triple during the **Reasoning** process. And the LLM succeed in re-ordering the whole candidate list based on the contexts of candidates and giving the correct answer during the **Re-ranking** process.

KG Triplet for completion: ([MASK], */location/adjoining_relationship/adjoins*, Champaign)

Task for completion: "The question is to predict the head entity [MASK] from the given ([MASK], *location adjoining_relationship adjoins*, Champaign) by completing the sentence 'Champaign is the adjoins of what location? The answer is '."

Re-Ranking:

The question is to predict the head entity [MASK] from the given ([MASK], *location adjoining_relationship adjoins*, Champaign) by completing the sentence 'Champaign is the adjoins of what location? The answer is '. The list of candidate answers is [Cook County, Champaign County, Bloomington, Evanston, Urbana].
Sort the list to let the candidate answers which are more possible to be the true answer to the question prior. Output the sorted order of candidate answers using the format '[most possible answer, second possible answer, ..., least possible answer]' and please start your response with 'The final order:'.

Context-aware Re-Ranking:

Champaign: city in Champaign County, Illinois, United States
The question is to predict the head entity [MASK] from the given ([MASK], *location adjoining_relationship adjoins*, Champaign) by completing the sentence 'Champaign is the adjoins of what location? The answer is '. The list of candidate answers is [Cook County, Champaign County, Bloomington, Evanston, Urbana].
Cook County: county in Illinois, United States
Champaign County: county in Illinois, United States
Bloomington: city and the county seat of McLean County, Illinois, United States
McHenry County: county in Illinois, United States
Evanston: suburban city in Cook County, Illinois, United States
Urbana: town in and county seat of Champaign County, Illinois, United States
Sort the list to let the candidate answers which are more possible to be the true answer to the question prior. Output the sorted order of candidate answers using the format '[most possible answer, second possible answer, ..., least possible answer]' and please start your response with 'The final order:'.

Re-Ranking Result generated by LLM:

The final order: [Urbana, Champaign County, Cook County, Bloomington, McHenry County Evanston]

Evaluation: The ground truth 'Urbana' hits at 1

Table 9: Prompt Template of context-aware ranking.