

# Research Trends for the Interplay between Large Language Models and Knowledge Graphs

Hanieh Khorashadizadeh  
University of Lübeck  
Lübeck, Germany  
hanieh.khorashadizadeh@uni-luebeck.de

Frédéric Ieng  
Université Paris Cité  
Paris, France  
frederic.ieng@u-paris.fr

Jinghua Groppe  
University of Lübeck  
Lübeck, Germany  
jinghua.groppe@uni-luebeck.de

Fatima Zahra Amara  
University of Khenchela  
Khenchela, Algeria  
f.amara@univ-khenchela.dz

Sanju Tiwari  
Alliance University  
Bangalore, India  
tiwarisanju18@ieee.org

Soror Sahri  
Université Paris Cité  
Paris, France  
soror.sahri@parisdescartes.fr

Sven Groppe  
University of Lübeck  
Lübeck, Germany  
sven.groppe@uni-luebeck.de

Morteza Ezzabady  
Université de Toulouse  
Toulouse, France  
morteza.ezzabady@irit.fr

Nandana Mihindukulasooriya  
IBM Research  
New York, US  
nandana@ibm.com

Farah Benamara  
Université de Toulouse  
Toulouse, France  
farah.benamara@irit.fr

## ABSTRACT

This survey investigates the synergistic relationship between Large Language Models (LLMs) and Knowledge Graphs (KGs), which is crucial for advancing AI's understanding, reasoning, and language processing capabilities. It aims to address gaps in current research by exploring areas such as KG Question Answering, ontology generation, KG validation, and enhancing KG accuracy and consistency through LLMs. The paper further examines the roles of LLMs in generating descriptive texts and natural language queries for KGs. Through a structured analysis that includes categorizing LLM-KG interactions, examining methodologies, and investigating collaborative uses and potential biases, this study seeks to provide new insights into the combined potential of LLMs and KGs. It highlights the importance of their interaction for improving AI applications and outlines future research directions.

## VLDB Workshop Reference Format:

Hanieh Khorashadizadeh, Fatima Zahra Amara, Morteza Ezzabady, Frédéric Ieng, Sanju Tiwari, Nandana Mihindukulasooriya, Jinghua Groppe, Soror Sahri, Farah Benamara, and Sven Groppe. Research Trends for the Interplay between Large Language Models and Knowledge Graphs. VLDB 2024 Workshop: LLM+KG.

This work is licensed under the Creative Commons BY-NC-ND 4.0 International License. Visit <https://creativecommons.org/licenses/by-nc-nd/4.0/> to view a copy of this license. For any use beyond this license, obtain permission by emailing [info@vldb.org](mailto:info@vldb.org). Copyright is held by the owner/author(s). Publication rights licensed to the VLDB Endowment.  
Proceedings of the VLDB Endowment. ISSN 2150-8097.

## 1 INTRODUCTION

In the era of artificial intelligence, Large Language Models (LLMs) and Knowledge Graphs (KGs) stand out as two pivotal technologies driving advancements in machine understanding, reasoning, and natural language processing. LLMs, such as OpenAI's GPT series [7], have demonstrated remarkable capabilities in generating human-like text, answering questions, and even creating content across various domains [46]. On the other hand, KGs organize and integrate information in a structured format, enabling machines to understand and infer relationships between entities in the real world [88]. The synergy between LLMs and KGs opens new frontiers for AI applications, enhancing machines' ability to process, interpret, and generate information with context and accuracy. Our intention with this survey is to fill the gaps left by previous survey papers [41, 67, 68, 90], offering fresh perspectives and uncovering new frontiers in the interaction between LLMs and KGs. For instance, our analysis delves deeper into KG Question Answering, encompassing multi-hop question answering, multi-hop question generation, and KG-powered chatbots. Additionally, our study extends to KG validation, an area not extensively discussed in previous survey papers. Table 1 outlines the categories discussed in previous research papers and this survey paper. Given the transformative potential and challenges of integrating LLMs with KGs, this paper seeks to explore the following research questions:

1. How can LLMs generate descriptive textual information for entities in a KG?
2. How can we employ LLMs in ontology generation?
3. How can LLMs help in detecting inconsistencies within KGs?

4. How can LLMs improve the accuracy, consistency, and completeness of KGs through fact-checking?
5. How can LLMs contribute to providing accurate answers for KG Question Answering?
6. How can LLMs effectively generate queries from natural language texts? (Text to Sparql or Cypher)

The interplay between LLMs and KGs can be categorized into three distinct types. The first, LLMs for KGs, demonstrates how LLMs are utilized to bolster and refine the functionality of KGs. The second, KG-enhanced LLMs, shows the inverse, where KGs are leveraged to enhance and inform the capabilities of LLMs. Lastly, LLM-KG Cooperation is depicted, representing a collaborative and synergistic relationship where LLMs and KGs work together to achieve more advanced and complex outcomes. Figure 1 illustrates the categorization of the interaction between LLMs and KGs. The remainder of this paper is structured to include an in-depth exploration of the three distinct types along with their respective sub-categories. The primary focus of this research is to address our formulated research questions. These specific areas of inquiry are highlighted in Figure 1, distinctly marked in pink, mentioned as Tasks Considered in our Research Questions. Those sections highlighted with stars represent topics that were not addressed in previous survey papers. The paper is organized as follows: Section 2 explores the role of LLMs within the context of KGs and Section 3 investigates how KGs can enhance LLMs. In Section 4, we examine the collaborative utilization of LLMs and KGs. Section 5 entails a statistics analysis and the identification of open challenges. Finally, we conclude in Section 6.

## 2 LLM FOR KG

### 2.1 KG Construction

**2.1.1 Ontology Creation.** This section of the paper explicitly addresses Research Question 2, delving into the application of LLMs in ontology generation for KGs. LLMs exhibit significant promise in constructing ontologies [64], where the organized representation of knowledge is essential for facilitating intelligent information retrieval and reasoning. LLMs contribute to creating and enriching comprehensive ontologies, with domain-specific fine-tuning enhancing their integration into specialized areas. While constructing a KG, [28] built an ontology for COVID-19 and demonstrated the effectiveness of LLMs. LLMs are promising for ontology generation but face computational intensity and bias challenges. They require broader evaluation yet remain valuable in ontology creation and intelligent systems.

*Concept and Relation Extraction:* LLMs like GPT-3, with their natural language understanding abilities, are suitable for extracting ontology concepts and relations. Concept extraction is the process of identifying and extracting concepts from text. The process involves extracting linguistic realizations of domain-specific concepts and accumulating semantic term variations with sense disambiguation and domain-specific synonym identification [73]. Relation extraction aims to identify non-taxonomic relations between concepts [4]. It seeks to identify relations between text entities and can be enhanced by using techniques like Named Entity Recognition (NER) and additional context provided by LLMs [11, 12].

*Property Identification:* Contributions like [76] investigate using fine-tuned LLMs for pre-annotating data in ontology property identification, helping annotators efficiently discern relevant properties and their correlation with automated scores, thereby reducing annotation times.

*Ontology Enrichment:* LLMs can help to improve current ontologies by adding new information or improving concept definitions. This is especially beneficial when working with dynamic domains in which the ontology must adapt to new facts. Advanced machine learning methods, remarkably fine-tuned LLMs, have been researched for data pre-annotation and ontology enrichment [76].

*Ontology Alignment:* LLMs can help to align ontologies by understanding and comparing the semantics of ideas and relationships among ontologies. Ontology alignment with LLMs has been studied in the literature. One method suggests a neurosymbolic design that combines the flexibility of LLMs and the domain orientation of Enterprise KGs (EKGs) [6]. Another work investigates the use of fine-tuned LLMs to pre-annotate data in a lexical extension job, specifically adding descriptive words to an existing ontology [76].

*Text-to-Ontology Mapping:* Train LLMs to map natural language text to specific ontology concepts and relationships. This can be accomplished using supervised learning, which involves training the model using annotated instances of text-to-ontology mappings. Text-to-ontology mapping using LLMs has been explored in multiple papers. One approach involves using artificial neural networks and classifiers to match texts to relevant ontologies [50].

*Ontology Learning:* Ontology learning automates the construction of ontologies by identifying concepts from text, aiming for a cost-effective, scalable knowledge representation across disciplines. Recent advances in LLMs, such as BERT, offer a promising alternative to traditional, time-consuming ontology creation, outperforming classic word embedding models in various nlp tasks and providing a more efficient method for knowledge acquisition in ontology development [4, 16, 29, 73].

**2.1.2 Entity Extraction and Alignment:** The Named Entity Recognition (NER) task involves being provided with a sentence, and the objective is to predict the set of entities present within that sentence[3]. Some papers concentrated on prompting to extract entities. Ashok and Lipton approach named PromptNER, comprises four essential components: a backbone LLM, a prompt outlining the set of entity types, a small set of examples from the target domain, and a specific format for outputting the extracted entities. Some approaches employ instruction-tuning. It is a technique in which pre-trained autoregressive language models undergo finetuning to adhere to natural language instructions and produce responses [66]. Zhou et al. proposed a focused distillation method with instruction tuning tailored to a specific mission.

**2.1.3 Relation Extraction:** LLMs have notably advanced the complex task of relation extraction between text entities, revolutionizing the identification and interpretation of semantic connections in textual data. In particular, GPT-RE [79] introduced a novel approach by infusing task-specific knowledge and logic-enriched demonstrations. This approach effectively reduces the low relevance issues between entity-relation pairings and overcomes the limitations in

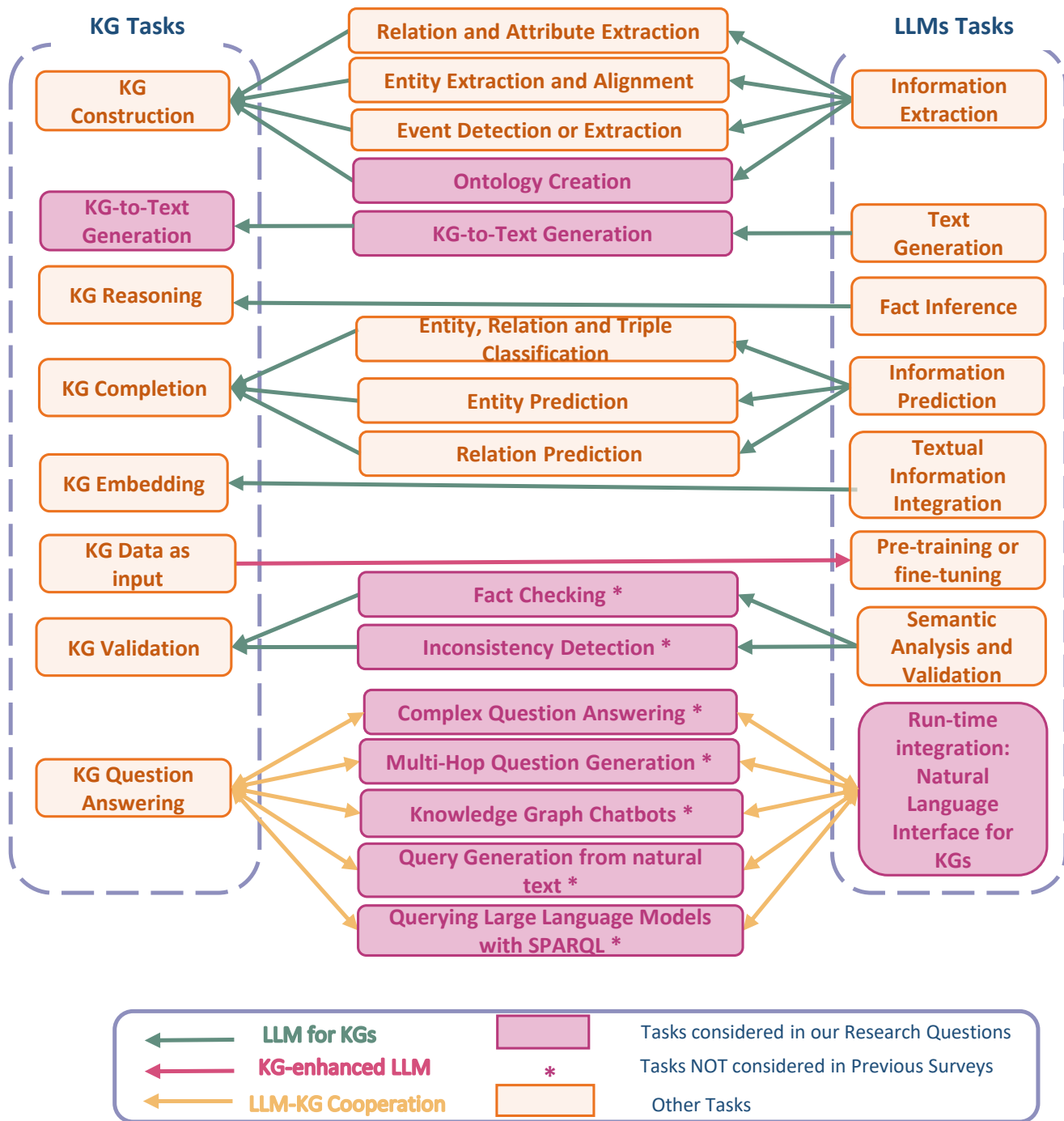


Figure 1: Categorization of the interplay between LLMs and KGs

Table 1: Categorizations addressed by previous survey papers

Main Category	Subcategory	Survey: [68]	Survey: [67]	Survey: [41]	Survey: [90]	Our Survey
KG Construction	Relation and Attribute Extraction	✓	✓	✗	✗	✓
	Entity Extraction and Alignment	✓	✓	✗	✗	✓
	Event Detection or Extraction	✗	✗	✗	✗	✗
	Ontology Creation	✗	✓	✗	✗	✓
KG-to-Text Generation	KG-to-Text Generation	✓	✗	✗	✗	✓
KG Reasoning	KG Reasoning	✓	✓	✗	✗	✓
KG Completion	Entity, Relation and Triple Classification	✓	✓	✗	✗	✓
	Entity Prediction	✓	✓	✗	✗	✓
	Relation Prediction	✗	✓	✗	✗	✓
KG Embedding	KG Embedding	✓	✗	✗	✗	✓
KG-enhanced LLM	KG-enhanced LLM	✓	✓	✓	✓	✓
KG Validation	Fact Checking	✗	✗	✗	✗	✓
	Inconsistency Detection	✗	✗	✗	✗	✓
KG Question Answering	Complex Question Answering	✗	✗	✗	✗	✓
	Multi-Hop Question Generation	✗	✗	✗	✗	✓
	Knowledge Graph Chatbots	✗	✗	✗	✗	✓
	Query Generation from natural text	✗	✗	✗	✗	✓
	Querying Large Language Models with SPARQL	✗	✗	✗	✗	✓

explaining input-label correlations. Furthermore, considering the vast array of predefined relation types and the challenges posed by uncontrolled LLMs, the study [55] suggests incorporating a natural language inference module within LLMs. This integration focuses on generating relation triples to greatly improve the usefulness and precision of document-level relation datasets. These advancements in relation extraction demonstrate LLMs’ capability to improve semantic analysis in text, and we have organized these methods by their distinct learning approaches. This classification comprises:

*Supervised Fine-tuning:* Supervised fine-tuning known for being prevalent and promising, involves extensively inputting training data to fine-tune LLMs. It allows LLMs to learn and comprehend the structural patterns in data, enhancing their ability to generalize. The Deepstruct model, referenced by [81], uses structure pre-training on diverse corpora to improve language models’ understanding of complex language structures, leading to a more nuanced interpretation of linguistic patterns. Huguet Cabot and Navigli enhanced relation extraction by training a specialized BART-style model [53] with a novel triplet linearization technique, which demonstrates the importance of supervised fine-tuning in advancing LLMs for complex tasks like relation extraction.

*Few-shot learning:* Few-shot learning, which relies on a limited number of labeled examples, faces significant challenges in machine learning. The lack of data can cause problems like overfitting and difficulty in understanding complex data relationships, as pointed out by [42]. The advent of LLMs has been a significant breakthrough in overcoming few-shot learning challenges. Xu et al. introduced two strategies for using LLMs in relation extraction: 1) in-context learning (ICL) and 2) data generation. ICL involves using detailed prompts with task definitions and labels to help LLMs understand relation extraction nuances, which is particularly useful in few-shot

settings with limited data. To further address data scarcity, they used LLMs to generate additional training material by creating prompts with instance descriptions and examples. In a complementary study, [78] examined the effectiveness of LLMs in few-shot relation extraction through in-context learning. They discovered that LLMs could equal state-of-the-art techniques using just a few examples. Their method included enhancing relation extraction labels with Chain of Thought (CoT) explanations from GPT-3 and fine-tuning the Flan-T5 model with this enriched data.

*Zero-shot learning:* Zero-shot learning aims to empower models to generalize to new tasks and domains beyond their training by adapting the pre-training of LLMs to unfamiliar situations. The extensive knowledge embedded in LLMs demonstrates their remarkable capability for zero-shot tasks in various fields. Key studies such as those by [49, 85], and recent advancements [54], exemplify the potential of LLMs in adapting to tasks they were not explicitly trained for. Adding to this discourse, [94] critically assessed ChatGPT’s zero-shot learning capability in temporal relation extraction. The study emphasized ChatGPT’s skill in grasping complex temporal relations but also noted its limitations in consistency and handling long-dependency relations.

## 2.2 KG-to-Text Generation:

This segment of the paper is dedicated to addressing Research Question 1, which investigates the effective use of LLMs in generating descriptive textual information for entities within KGs. KG-to-Text generation encompasses the procedure of transforming structured data held within a KG into human-readable, natural-language text [22]. Colas et al. combined a pre-trained LM with graph attention information by modifying the encoder layer. Initially, they convert the KG into a textual string by linearizing its structure.

Then, the vector representation becomes contextualized through the incorporation of graph attention data. Several methods involve the fine-tuning of pre-trained language models (PLM), specifically on KG-to-text datasets, to address this particular task [70]. Chen et al. gathered a corpus named KGTEXT, which involves matching linked sentences sourced from Wikipedia with knowledge sub-graphs obtained from WikiData and then conducting pre-training and subsequently fine-tuning on KG-to-Text datasets. Ke et al. developed a basic yet effective structure-aware semantic aggregation module that integrates into every Transformer layer within the encoder. The purpose of this module is to maintain the structural details present in the input graphs. Then followed by three pre-training tasks aimed at explicitly acquiring graph-text alignments. Li et al. investigated few-shot KG-to-text generation, which generates the output with the help of around several hundred instances. Initially, their model aligns KG representations, encoded by graph neural networks, PLM-based entity representations, aiming to close the semantic gap. It then proposes a specialized strategy, called relation-biased breadth-first search (RBFS), to arrange the KG into a well-structured entity sequence for integration into PLMs. Lastly, the model employs multi-task learning, simultaneously training the main text generation and an auxiliary KG reconstruction task. This approach significantly improves the semantic alignment between the input KG and the resulting text, allowing the model to generate accurate descriptions of the KG.

### 2.3 KG Reasoning

KG Reasoning involves deriving new facts from existing ones using logical inference. Recently, there has been a growing emphasis on exploring KG reasoning due to its potential to derive new knowledge and conclusions from existing data [18]. It has applications for KG completion and KG question answering [18]. Choudhary and Reddy developed Language-guided Abstract Reasoning over Knowledge graphs (LARK) model that is designed to harness the reasoning power of LLMs for efficiently answering first-order logic (FOL) queries on knowledge graphs. LARK identifies relevant sub-graph contexts within abstract KGs by utilizing the entities and relationships in the queries. It then conducts chain reasoning over these contexts using prompts from LLMs to break down and process logical queries. Luo et al. introduced Reasoning on Graphs (RoG), which combines LLMs and KGs to perform reliable and easily understandable reasoning. They proposed a framework for planning-retrieval-reasoning to tackle challenges such as hallucinations and knowledge deficiencies. Initially, the planning module produces relation paths in the form of faithful plans using KGs, forming the basis for the RoG approach. Subsequently, these devised plans are employed to extract legitimate paths of reasoning from KGs to carry out reliable reasoning through the retrieval-reasoning module. Kim et al. introduced KG-GPT, a versatile framework for reasoning on KG that operates in three stages: Sentence Segmentation, Graph Retrieval, and Inference. These stages are designed to break down sentences, extract relevant components from the graph, and draw logical conclusions.

### 2.4 KG Completion

In KG completion, the main tasks are (i) triple classification, validating the accuracy of a given triple; (ii) link prediction for predicting missing components like the head, tail, or relation in a triple. And (iii) entity classification for categorizing entities, also known as determining the entity’s type [2]. The triple-based methods relying on structural information are widely used for KG completion, mostly the embedding methods [9, 58, 77]. Several works have attempted to prove more effectiveness than triple-based completion by extracting factual knowledge from LMs embedding spaces instead of simple structures. Examples of these text-based methods are KG-BERT [92] and the fine-tuning of GPT-2 [8]. KG-BERT [92] is a leading example of applying LLMs to KG completion. It fine-tunes for this task by treating triples as textual sequences. The model uses KG-BERT to score sequences of entity-relation-entity, representing these elements through their names or descriptions. These sequences of words are then used as input for fine-tuning the BERT model. KG-BERT was identified with two shortcomings: 1) the model’s struggle to learn relational data in KGs, and 2) difficulty in identifying the correct answers among similar candidates. To improve this, multi-task learning methods combining relation prediction and relevance ranking with link prediction were proposed, as in [47]. This enhanced model better grasps KG relationships and effectively handles lexical similarities. In another study, [80] introduced the Structure-augmented Text Representation (StAR) model aimed at improving link prediction in KG completion. The model begins with a Siamese-style textual encoder applied to triples, generating two contextualized representations. A distinctive scoring module using dual strategies captures both contextualized and structured knowledge, enhanced by a self-adaptive ensemble scheme with graph embedding techniques to boost performance. Wang et al. presented SimKGC, a straightforward approach to enhance text-based KG completion. It recognizes the crucial aspect of conducting efficient contrastive learning. Building on recent advancements in contrastive learning, SimKGC utilizes a bi-encoder architecture and incorporates three types of negative examples to achieve its objectives. Sequence-to-sequence PLM methods have also shown state-of-the-art performance by redefining the KG completion challenge as a sequence-to-sequence generation task as demonstrated in KG-S2S [15] and GenKGC [87]. GenKGC incorporates innovative techniques, such as relation-guided demonstration and entity-aware hierarchical decoding, aimed at improving the quality of representation learning and accelerating the inference process. Most of these recent text-based methods require specific fine-tuning for KGs, which limits their efficiency. New training-free methods with no fine-tuning were recently proposed to address this limitation, as in KICGPT [86].

### 2.5 KG Embedding

KG Embedding involves mapping elements of a KG into vectors in a continuous vector space. Unlike Graph Neural Networks, which learn KG embeddings from data explicitly represented by nodes and edges, LM learns KG embeddings from natural language. Given a piece of text, LMs learn representations of entities, which capture context information containing relations of entities. The entity relations are found by an attention mechanism, which can create

an association between two entities independent of their location. The more text an LM sees the more entities and their relationships the LM can learn; the larger an LM, the more contextual information the representation contains. LLMs contain billions of parameters and are trained over a huge amount of textual data (e.g. 45 TB for GPT-3). LLMs perfectly meet these requirements, and so are a powerful tool for learning KG embeddings. LLMs’ ability to KG embedding attracts researchers to use them directly to conduct KG tasks, like entity alignment [59]. We can also use the representation of entities learned by LLMs in the small-sized models, and this should significantly reduce the amount of training data needed and the time of training and, meanwhile, create well-generalized models. An extensive experiment is needed to investigate the efficiency of applying embeddings of LLMs into small-sized models for KG analysis tasks.

## 2.6 KG Validation

**2.6.1 Fact Checking.** In this section, we address Research Question 4, which is centered on how LLMs can enhance the accuracy, consistency, and completeness of KGs through the process of fact-checking. The advent of LLMs has the potential to significantly impact the fight against misinformation, presenting both opportunities and challenges. LLMs, with their extensive world knowledge and strong reasoning abilities, offer promise in combating misinformation. However, the flip side is that these models can also be easily manipulated to generate deceptive misinformation at scale [14]. An approach to address misinformation in a KG involves verbalizing each triple within the KG and prompting LLMs to identify instances of misinformation [7, 13]. To address the limitations of potentially outdated or insufficient knowledge in LLMs for detecting factual errors, some research has explored augmenting LLMs with external knowledge [20] or utilizing tools [19] for misinformation detection. The methodologies developed for factual checking in LLMs can be applied to KG fact-checking by converting KG triples into textual representations.

**2.6.2 Inconsistency Detection.** This section delves into Research Question 3, examining the role of LLMs in identifying and fixing inconsistencies in KGs. KG consistency is about ensuring that the information in a KG is not self-contradictory [95]. KG consistency differs from KG accuracy in that it prioritizes the logical soundness of the KG rather than the factual correctness of individual triples. A KG might contain outdated yet logically coherent information, maintaining high consistency even with low accuracy. Conversely, high accuracy usually implies good consistency, as factual data tends to be logically harmonious [83]. Current rule-based techniques for KG completion and inconsistency detection primarily utilize structural information, overlooking the semantics of relations. Using LLMs can harness both the semantic and structural information from KGs to generate meaningful rules. Among recent works leveraging LLMs for rule-based methods, there is ChatRule [61], a framework that uses both the semantic and structural information of KGs to prompt LLMs to generate logical rules.

## 3 KG-ENHANCED LLM

K-BERT [60] proposed to inject relevant domain knowledge from a KG into a sentence using a knowledge layer before the embeddings

are calculated. The authors show that it improves performance in many NLP tasks. Sem-K-BERT [88] further extended this by integrating semantic role labeling (SRL) and by introducing semantic correlation calculation to reduce the noise. KCF-NET [31] demonstrated that incorporating KG representations with context helped to improve the LM performance in machine reading comprehension. Ji et al. proposes a concept-enhanced pertaining approach by adding concept semantics to sentences from external sources such as KGs. Dict-BERT [93] leverages definitions of rare words from dictionaries to improve the performance of various NLP tasks. Retrieval Augmented Generation (RAG) was developed to address issues such as hallucination, insufficient domain-specific knowledge, and outdated information by incorporating relevant external knowledge into the LLM prompt [30]. RAG has three steps: Retrieval, Generation, and Augmentation. Over time, RAG has developed into three distinct models: Naive RAG, Advanced RAG, and Modular RAG. Naive RAG operates through a three-step process: indexing, retrieval, and generation. During the indexing step, text data is divided into small segments, each of which is encoded into vector form using embedding models. In the retrieval phase, the user’s query is similarly converted into a vector. The system then computes the similarity between the query vector and the vectors of the text segments, selecting the top k segments as the context for the response. Finally, in the generation step, both the query and the chosen contexts are input into an LLM as a prompt to generate a response. The Modular RAG has the ability to retrieve pertinent information from knowledge graphs[30]. [84] introduced KnowledgeGPT, a system that enhances LLMs with data from knowledge bases. To respond to a user query, their method generates a search code and executes it to retrieve information from the knowledge base. The output is then given to the LLM to respond to the user query. While RAG is effective in responding to queries directly related to specific excerpts from the source data where the retrieval is explicit, it may struggle with more general queries. For example, it might not perform well when asked, "What are the main points of the dataset?" To overcome this limitation, Graph RAG was introduced [26]. This approach generates a summary over a knowledge graph that has been constructed by the LLM from the source data.

## 4 LLM-KG COOPERATION

### 4.1 KG Question Answering

This section covers research questions 5 and 6.

**4.1.1 Multi-Hop Question Generation:** The goal of multi-hop question generation is to formulate questions demanding advanced reasoning over multiple sentences, including their associated answers [57]. Li et al. created a KG Enhanced Language Model (KGEL), aiming to replicate human-like reasoning for multi-hop questions. Their method for generating multi-hop questions reflects the way humans approach complex questioning. Their strategy encompasses three key phases: (1) Gaining an understanding of the context by employing a pre-trained GPT-2 language model, (2) Merging information and reasoning through a sophisticated KG and an answer-aware dynamic Graph Attention Network (GAT), and (3) Executing the question generation using a multi-head attention generation module that capitalizes on improved latent representations. Aigo et al.

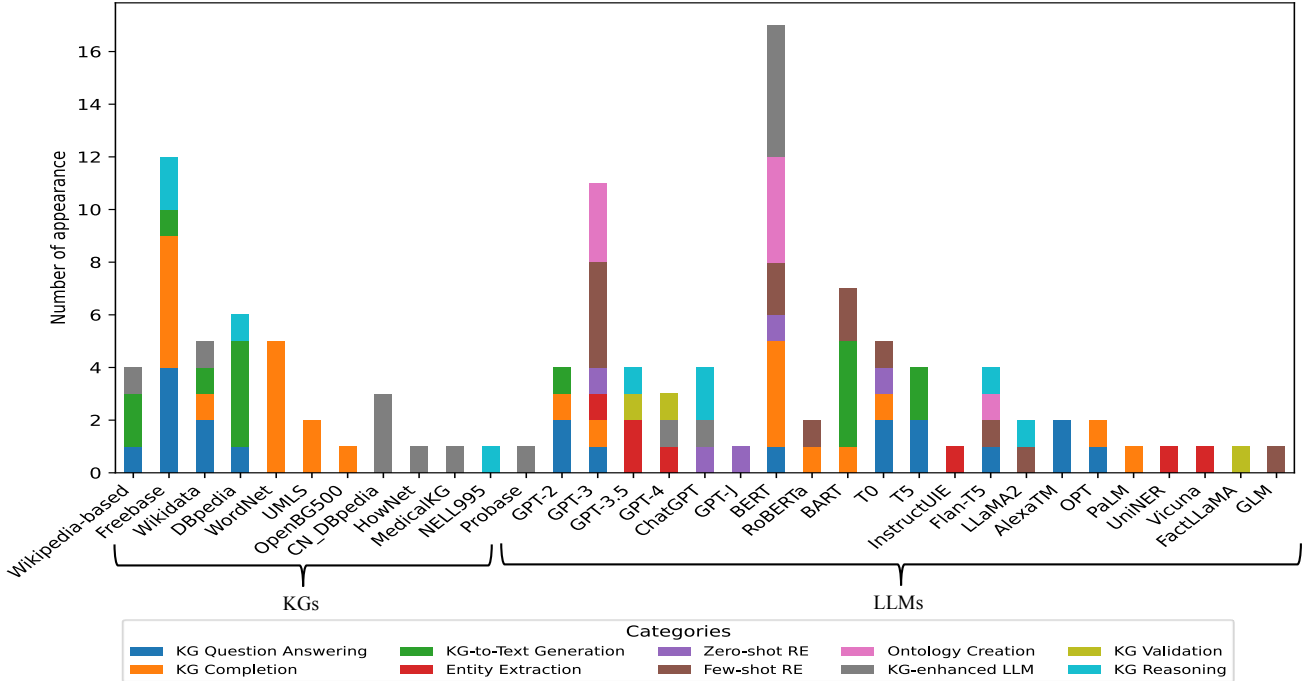


Figure 2: Statistics of the usage of LLMs and KGs in cited papers per category

research primary emphasis revolved around formulating inquiries by employing KGs in conjunction with the T5 language model. They harnessed the language model’s capabilities to generate questions while also applying a technique to obstruct self-attention in the encoder, all to train the model with a deliberate emphasis on maintaining the inherent structure of the graph. They harnessed the language model’s capabilities to generate questions while also applying a technique to obstruct self-attention in the encoder, all to train the model with a deliberate emphasis on maintaining the inherent structure of the graph. Their research didn’t target multi-hop question generation.

**4.1.2 Complex or Multi-hop Question Answering:** Complex question answering over KGs involves multiple subjects, conveys interconnected relations, includes numerical operations [52], and demands multi-hop reasoning [27, 91]. Cao and Liu introduced a novel method, ReLMKG, designed for the multi-hop Knowledge Base Question Answering (KBQA) task. The approach leverages both the implicit knowledge embedded in a pre-trained language model and the explicit knowledge from KGs. They transferred the KG into a textual format, aligning it with the question in a shared space. The textualized graph and the question are input for the pre-trained language model, which acts as the question-path encoder. The lack of the inherent topological arrangement in the textual KG, which holds essential explicit knowledge for model reasoning, led to the creation of a path-centric Graph Neural Network (GNN) as the reasoning element. The instructions for the reasoning module are derived from the outputs of the pre-trained language

model. Sen et al. introduced an innovative approach to address queries, employing a KGQA model to extract information from a KG. Simultaneously, a language model is employed to analyze the question and facts, enabling the derivation of a well-reasoned answer. Baek et al. utilized a similarity metric between KG facts and questions. This approach was employed to retrieve pertinent facts, subsequently enhancing the prompt of a language model.

**4.1.3 Query Generation from text:** This portion of the paper specifically addresses Research Question 6, which examines the effectiveness of LLMs in generating natural language queries. It particularly focuses on converting textual descriptions into query languages like SPARQL or Cypher for KGs. Constructing SPARQL queries from natural language questions is challenging, as it demands an understanding of both the question itself and the underlying patterns within the KG [71]. Rony et al. introduced a novel embedding technique called SGPT, tailored to encode linguistic features from the question, and suggested training methods that make use of a pre-trained language model to produce a SPARQL query. Kovriguina et al. proposed a one-shot generative approach, termed SPARQL-GEN, which aims to generate SPARQL queries by augmenting LLMs with pertinent context provided in a single prompt. This prompt encompasses the question, an RDF subgraph necessary for answering the question, and an example of a correct SPARQL query for a different question. Pliukhin et al. devised a method that incorporates diverse data sources in the SPARQL generation prompt, encompassing the question, an RDF subgraph essential for answering the question, and an illustration of a correct SPARQL query. Their

method suggests a couple of enhancements to the SPARQLGEN approach.

**4.1.4 Querying LLMs with SPARQL:** Saeed et al. explored the possibility of querying LLMs with SQL, initiating new directions for research exploration. While SQL is a powerful tool for structured datasets with defined schemas, it falls short in immediately handling information presented as unstructured text. The statement discusses an illustration of a Database-First (DB-first) approach that harnesses the capabilities of LLMs along with traditional Database Management Systems (DBMSs). This combination is utilized to establish a hybrid query execution environment, suggesting an integration of LLMs with conventional database systems for enhanced performance. This concept might also align with querying LLMs with SPARQL, which might obtain hidden relations in unstructured data.

**4.1.5 KG Chatbots:** Omar et al. conducted a detailed examination and comparison between conversational AI models like ChatGPT and traditional Question-Answering Systems (QASs) designed for KGs and suggested merging the attributes of both with the goal of creating KG Chatbots.

## 5 STATISTICS ANALYSIS AND OPEN CHALLENGES

### 5.1 Statistics about used LLMs and KGs in Approaches

Our analysis categorizes research papers based on the use of LLMs and KGs. Figure 2 illustrates this, with the y-axis representing the frequency of appearances and the x-axis listing the specific KGs and LLMs. The data reveal that Freebase is the most commonly utilized KG in the reviewed literature. Regarding LLMs, Bert and GPT-3 emerge as the most frequently employed models.

### 5.2 Open Challenges

We still need research to find techniques to incorporate knowledge reliably into the answers of LLMs and to go for smaller-sized LLMs without losing the capabilities of LLMs to reason on their input. One way seems to be to incorporate the knowledge from KGs reliably into the inference process of LLMs and to exclude the knowledge from the training data because these facts are then not needed anymore to be stored in the neural network of the LLMs [63] increasing its number of parameters. Also, incorporating only code of a core fragment of a programming language or a query language like SPARQL [35] and XPath [33, 37] without redundant language constructs in the training data may reduce the number of needed parameters in LLMs. To reduce the number of needed parameters, future work may also consider generating code and queries in a limited set of programming and query languages and transforming the generated code to the requested programming and query languages. For example, automated transformations between XML and Semantic Web data format and query languages are well known [23–25, 38, 39], and also reformulating queries based on a given transformation of the data [36]. Other techniques may consider only those queries in the training data, which can return a result and are hence satisfiable [32–34, 40]. Running LLMs with

fewer parameters reduces the energy needed and, hence, the carbon footprint of running LLMs. We see a demand for a better separation of knowledge (to be managed in KGs in a condensed and reliable way) and natural language understanding (based on a well-chosen minimal training data set of high quality not addressing the knowledge of the given KG). How to retrieve this well-chosen minimal training data set is a non-trivial question and may open up a new research direction itself. With a clear separation of knowledge and natural language understanding, fine-tuning to specific domains may also become obsolete or at least can be reduced to a minimum effort. New applications of LLMs like Personal KG-enhanced LLMs, which can imitate the style of writing of each individual by fine-tuning from email and chat conversations and based on a Personal KG containing the (private) knowledge of the individual might become true.

Besides reducing the complexity of LLMs there might be another research direction on more complex architectures mimicking the working of the human brain even more towards the holy grail of artificial intelligence, the artificial general intelligence (AGI), to accomplish any intellectual task that human beings or animals can perform [75]. In these architectures, LLMs or advanced variants might play only a minor role by being responsible for processing the input and/or verbalizing the conclusion of the overall process. KGs might still be a standard way for administrating knowledge in a reliable and space-efficient way in these architectures for AGI.

## 6 CONCLUSION

In conclusion, this comprehensive survey has significantly contributed to the understanding and advancement of the integration between LLMs and KGs. Through our meticulous exploration across various sections, we have addressed critical research questions, shedding light on how LLMs can enhance KGs in areas such as generating descriptive textual information, ontology generation, inconsistency detection, fact-checking, KG completion, and KG embedding. Our study also investigates how KGs can improve LLMs and explores the cooperation between LLMs and KGs. Specifically, it examines under-discussed areas in KG question answering, including query generation from text and multi-hop question generation and answering.

## ACKNOWLEDGMENTS

The German Research Foundation funds this work under project number 490998901.

## REFERENCES

- [1] Kosuke Aigo, Takashi Tsunakawa, Masafumi Nishida, and Masafumi Nishimura. 2021. Question generation using knowledge graphs with the T5 language model and masked self-attention. In *GCCE*. IEEE.
- [2] Mehwish Alam, Frank van Harmelen, and Maribel Acosta. 2023. Towards Semantically Enriched Embeddings for Knowledge Graph Completion. *arXiv preprint arXiv:2308.00081* (2023).
- [3] Dhananjay Ashok and Zachary C Lipton. 2023. PromptNER: Prompting For Named Entity Recognition. *arXiv preprint arXiv:2305.15444* (2023).
- [4] Hamed Babaei Giglou, Jennifer D’Souza, and Sören Auer. 2023. LLMs4OL: Large language models for ontology learning. In *ISWC*.
- [5] Jinheon Baek, Alham Fikri Aji, and Amir Saffari. 2023. Knowledge-Augmented Language Model Prompting for Zero-Shot Knowledge Graph Question Answering. *arXiv preprint arXiv:2306.04136* (2023).
- [6] Teodoro Baldazzi, Luigi Bellomarini, Stefano Ceri, Andrea Colombo, Andrea Gentili, and Emanuel Sallinger. 2023. Fine-tuning Large Enterprise Language



- Models via Ontological Reasoning. *arXiv:2306.10723* (2023).
- [7] Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. A multitask, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023* (2023).
  - [8] Russa Biswas, Radina Sofronova, Mehwish Alam, and Harald Sack. 2021. Contextual Language Models for Knowledge Graph Completion. In *MLSMKG@PKDD/ECML*.
  - [9] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems* 26 (2013).
  - [10] Xing Cao and Yun Liu. 2023. ReLMKG: reasoning with pre-trained language models and knowledge graphs for complex question answering. *Applied Intelligence* 53, 10 (2023), 12032–12046.
  - [11] J Harry Caufield, Harshad Hegde, Vincent Emonet, Nomi L Harris, Marcin P Joachimiak, Nicolas Matentzoglou, HyeongSik Kim, Sierra AT Moxon, Justin T Reese, Melissa A Haendel, et al. 2023. Structured prompt interrogation and recursive extraction of semantics (SPIRES): A method for populating knowledge bases using zero-shot learning. *arXiv:2304.02711* (2023).
  - [12] Daniel T Chang. 2023. Concept-Oriented Deep Learning with Large Language Models. *arXiv preprint arXiv:2306.17089* (2023).
  - [13] Canyu Chen and Kai Shu. 2023. Can llm-generated misinformation be detected? *arXiv:2309.13788* (2023).
  - [14] Canyu Chen and Kai Shu. 2023. Combating misinformation in the age of llms: Opportunities and challenges. *arXiv preprint arXiv:2311.05656* (2023).
  - [15] Chen Chen, Yufei Wang, Bing Li, and Kwok-Yan Lam. 2022. Knowledge is flat: A seq2seq generative framework for various knowledge graph completion. *arXiv preprint arXiv:2209.07299* (2022).
  - [16] Jiaoyan Chen, Yuan He, Yuxia Geng, Ernesto Jiménez-Ruiz, Hang Dong, and Ian Horrocks. 2023. Contextual semantic embeddings for ontology subsumption prediction. *World Wide Web* (2023), 1–23.
  - [17] Wenhui Chen, Yu Su, Xifeng Yan, and William Yang Wang. 2020. KGPT: Knowledge-grounded pre-training for data-to-text generation. *arXiv preprint arXiv:2010.02307* (2020).
  - [18] Xiaojun Chen, Shengbin Jia, and Yang Xiang. 2020. A review: Knowledge reasoning over knowledge graph. *Expert Systems with Applications* 141 (2020), 112948.
  - [19] I Chern, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, Pengfei Liu, et al. 2023. FacTool: Factuality Detection in Generative AI—A Tool Augmented Framework for Multi-Task and Multi-Domain Scenarios. *arXiv preprint arXiv:2307.13528* (2023).
  - [20] Tsun-Hin Cheung and Kin-Man Lam. 2023. FactLLaMA: Optimizing Instruction-Following Language Models with External Knowledge for Automated Fact-Checking. In *APSIPA ASC*. IEEE.
  - [21] Nurendra Choudhary and Chandan K Reddy. 2023. Complex logical reasoning over knowledge graphs using large language models. *arXiv preprint arXiv:2305.01157* (2023).
  - [22] Anthony Colas, Mehrdad Alvandipour, and Daisy Zhe Wang. 2022. GAP: A graph-aware language model framework for knowledge graph-to-text generation. *arXiv preprint arXiv:2204.06674* (2022).
  - [23] Matthias Droop, Markus Flarer, Jinghua Groppe, Sven Groppe, Volker Linnemann, Jakob Pinggera, Florian Santner, Michael Schier, Felix Schöpf, Hannes Staffler, and Stefan Zugal. 2007. Translating XPath Queries into SPARQL Queries. In *On the Move to Meaningful Internet Systems 2007: OTM Workshops, OTM Confederated International Workshops and Posters, AWeSOME, CAMS, OTM Academy Doctoral Consortium, MONET, OnToContent, ORM, PerSys, PPN, RDDS, SWS, and SWS, Vilamoura, Portugal*. [https://doi.org/10.1007/978-3-540-76888-3\\_5](https://doi.org/10.1007/978-3-540-76888-3_5)
  - [24] Matthias Droop, Markus Flarer, Jinghua Groppe, Sven Groppe, Volker Linnemann, Jakob Pinggera, Florian Santner, Michael Schier, Felix Schöpf, Hannes Staffler, and Stefan Zugal. 2008. Bringing the XML and Semantic Web Worlds Closer: Transforming XML into RDF and Embedding XPath into SPARQL. In *10th International Conference on Enterprise Information Systems (ICEIS), Barcelona, Spain, Revised Selected Papers*. [https://doi.org/10.1007/978-3-642-00670-8\\_3](https://doi.org/10.1007/978-3-642-00670-8_3)
  - [25] Matthias Droop, Markus Flarer, Jinghua Groppe, Sven Groppe, Volker Linnemann, Jakob Pinggera, Florian Santner, Michael Schier, Felix Schöpf, Hannes Staffler, and Stefan Zugal. 2008. Embedding XPath Queries into SPARQL Queries. In *Proceedings of the Tenth International Conference on Enterprise Information Systems (ICEIS), Barcelona, Spain*.
  - [26] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130* (2024).
  - [27] Romina Etezadi and Mehrnosh Shamsfard. 2023. The state of the art in open domain complex question answering: a survey. *Applied Intelligence* 53, 4 (2023), 4124–4144.
  - [28] Morteza Kamaladdini Ezzabady, Frédéric Ieng, Hanieh Khorashadizadeh, Farah Benamara, Sven Groppe, and Soror Sahri. 2024. Towards Generating High-Quality Knowledge Graphs by Leveraging Large Language Models. In *The 29th Annual International Conference on Natural Language & Information Systems (NLDB 2024), Turin, Italy*.
  - [29] Maurice Funk, Simon Hosemann, Jean Christoph Jung, and Carsten Lutz. 2023. Towards Ontology Construction with Language Models. *arXiv preprint arXiv:2309.09898* (2023).
  - [30] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997* (2023).
  - [31] Peizhu Gong, Jin Liu, Yihe Yang, and Huihua He. 2020. Towards Knowledge Enhanced Language Model for Machine Reading Comprehension. *IEEE Access* 8 (2020), 224837–224851.
  - [32] Jinghua Groppe and Sven Groppe. 2006. A Prototype of a Schema-Based XPath Satisfiability Tester. In *17th International Conference on Database and Expert Systems Applications (DEXA), Kraków, Poland*. [https://doi.org/10.1007/11827405\\_10](https://doi.org/10.1007/11827405_10)
  - [33] Jinghua Groppe and Sven Groppe. 2006. Satisfiability-Test, Rewriting and Re-Formulating XPath queries and XSLT queries on XSLT views. *Data Knowl. Eng.* 57, 1 (2006), 64–110. <https://doi.org/10.1016/j.datak.2005.04.002>
  - [34] Sven Groppe and Stefan Böttcher, and Jinghua Groppe. 2006. XPath Query Simplification with regard to the Elimination of Intersect and Except Operators. In *Proceedings of the 22nd International Conference on Data Engineering Workshops (ICDEW), Atlanta, GA, USA*. <https://doi.org/10.1109/ICDEW.2006.165>
  - [35] Sven Groppe, Jinghua Groppe, Niklas Klein, Ralf Bettentrupp, Stefan Böttcher, and Le Gruenwald. 2011. Transforming XSLT stylesheets into XQuery expressions and vice versa. *Computer Languages, Systems & Structures* 37, 2 (2011), 76–111. <https://doi.org/10.1016/j.cl.2010.11.001>
  - [36] Sven Groppe, Jinghua Groppe, Volker Linnemann, Dirk Kukulenz, Nils Hoeller, and Christoph Reinke. 2008. Embedding SPARQL into XQuery/XSLT. In *Proceedings of the 2008 ACM Symposium on Applied Computing (SAC), Fortaleza, Ceara, Brazil*. <http://doi.acm.org/10.1145/1363686.1364228>
  - [37] Sven Groppe, Jana Neumann, and Volker Linnemann. 2009. SWOBE - embedding the semantic web languages RDF, SPARQL and SPARUL into java for guaranteeing type safety, for checking the satisfiability of queries and for the determination of query result types. In *Proceedings of the 2009 ACM Symposium on Applied Computing (SAC), Honolulu, Hawaii, USA*. <http://doi.acm.org/10.1145/1529282.1529561>
  - [38] Limmei Hu, Zeyi Liu, Ziwang Zhao, Lei Hou, Liqiang Nie, and Juanzi Li. 2023. A survey of knowledge enhanced pre-trained language models. *IEEE Transactions on Knowledge and Data Engineering* (2023).
  - [39] Jiaxin Huang, Chunyuan Li, Krishan Subudhi, Damien Jose, Shobana Balakrishnan, Weizhu Chen, Baolin Peng, Jianfeng Gao, and Jiawei Han. 2020. Few-shot named entity recognition: A comprehensive study. *arXiv:2012.14978* (2020).
  - [40] Pere-Lluís Huguet Cabot and Roberto Navigli. 2021. REBEL: Relation Extraction By End-to-end Language generation. In *EMNLP 2021*. <https://doi.org/10.18653/v1/2021.findings-emnlp.204>
  - [41] Zizheng Ji, Lin Dai, Jin Pang, and Tingting Shen. 2020. Leveraging Concept-Enhanced Pre-Training Model and Masked-Entity Language Model for Named Entity Disambiguation. *IEEE Access* 8 (2020).
  - [42] Pei Ke, Haozhe Ji, Yu Ran, Xin Cui, Liwei Wang, Linfeng Song, Xiaoyan Zhu, and Minlie Huang. 2021. Jointgt: Graph-text joint representation learning for text generation from knowledge graphs. *arXiv preprint arXiv:2106.10502* (2021).
  - [43] Hanieh Khorashadizadeh, Nandana Mihindukulasooriya, Sanju Tiwari, Jinghua Groppe, and Sven Groppe. 2023. Exploring In-Context Learning Capabilities of Foundation Models for Generating Knowledge Graphs from Text. *arXiv:2305.08804* [cs.CL]
  - [44] 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 COLING*.
  - [45] Jiho Kim, Yeonsu Kwon, Yohan Jo, and Edward Choi. 2023. Kg-gpt: A general framework for reasoning on knowledge graphs using large language models. *arXiv preprint arXiv:2310.11220* (2023).
  - [46] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large Language Models are Zero-Shot Reasoners. <http://arxiv.org/abs/2205.11916> arXiv:2205.11916 [cs].
  - [47] Lukáš Korel, Uladzislau Yorsh, Alexander S Behr, Norbert Kockmann, and Martin Holeňa. 2023. Text-to-Ontology Mapping via Natural Language Processing with Application to Search for Relevant Ontologies in Catalysis. *Computers* 12, 1

- (2023), 14.
- [51] Liubov Kovriguina, Roman Teucher, Daniil Radyush, and Dmitry Mourmrtsev. 2023. SPARQLGEN: One-Shot Prompt-based Approach for SPARQL Query Generation. (2023).
- [52] Yunshi Lan, Gaole He, Jinhao Jiang, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. A survey on complex knowledge base question answering: Methods, challenges and solutions. *arXiv:2105.11644* (2021).
- [53] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of ACL 2020*. Online.
- [54] Guozheng Li, Peng Wang, and Wenjun Ke. 2023. Revisiting Large Language Models as Zero-shot Relation Extractors. <http://arxiv.org/abs/2310.05028> arXiv:2310.05028 [cs].
- [55] Junpeng Li, Zixia Jia, and Zilong Zheng. 2023. Semi-automatic Data Enhancement for Document-Level Relation Extraction with Distant Supervision from Large Language Models. arXiv:2311.07314[cs].
- [56] Junyi Li, Tianyi Tang, Wayne Xin Zhao, Zhicheng Wei, Nicholas Jing Yuan, and Ji-Rong Wen. 2021. Few-shot knowledge graph-to-text generation with pretrained language models. *arXiv preprint arXiv:2106.01623* (2021).
- [57] Zhenping Li, Zhen Cao, Pengfei Li, Yong Zhong, and Shaobo Li. 2023. Multi-Hop Question Generation with Knowledge Graph-Enhanced Language Model. *Applied Sciences* 13, 9 (2023), 5765.
- [58] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the AAAI conference*.
- [59] Anna Sofia Lippolis, Antonis Klironomos, Daniela F Milon-Flores, Heng Zheng, Alexane Jouglar, Ebrahim Norouzi, and Aidan Hogan. 2023. Enhancing Entity Alignment Between Wikidata and ArtGraph using LLMs. In *SWODCH@ISWC*.
- [60] Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-bert: Enabling language representation with knowledge graph. In *Proceedings of the AAAI Conference*.
- [61] Linhao Luo, Jiaxin Ju, Bo Xiong, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2023. ChatRule: Mining Logical Rules with Large Language Models for Knowledge Graph Reasoning. *ArXiv abs/2309.01538* (2023).
- [62] Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2023. Reasoning on graphs: Faithful and interpretable large language model reasoning. *arXiv preprint arXiv:2310.01061* (2023).
- [63] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and Editing Factual Associations in GPT. In *Advances in Neural Information Processing Systems* 35.
- [64] Fabian Neuhaus. 2023. Ontologies in the era of large language models—a perspective. *Applied Ontology* 18, 4 (2023), 399–407.
- [65] Reham Omar, Omij Mangukiya, Panos Kalnis, and Essam Mansour. 2023. Chatgpt versus traditional question answering for knowledge graphs: Current status and future directions towards knowledge graph chatbots. *arXiv preprint arXiv:2302.06466* (2023).
- [66] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback, 2022. URL <https://arxiv.org/abs/2203.02155> 13 (2022).
- [67] Jeff Z Pan, Simon Razniewski, Jan-Christoph Kalo, Sneha Singhania, Jiaoyan Chen, Stefan Dietze, Hajira Jabeen, Janna Omelivanenko, Wen Zhang, Matteo Lissandrini, et al. 2023. Large language models and knowledge graphs: Opportunities and challenges. *arXiv preprint arXiv:2308.06374* (2023).
- [68] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE TKDE* (2024).
- [69] Dmitrii Plukhin, Daniil Radyush, Liubov Kovriguina, and Dmitry Mourmrtsev. 2023. Improving Subgraph Extraction Algorithms for One-Shot SPARQL Query Generation with Large Language Models. (2023).
- [70] Leonardo FR Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2020. Investigating pretrained language models for graph-to-text generation. *arXiv preprint arXiv:2007.08426* (2020).
- [71] Md Rashad Al Hasan Rony, Uttam Kumar, Roman Teucher, Liubov Kovriguina, and Jens Lehmann. 2022. SGPT: a generative approach for SPARQL query generation from natural language questions. *IEEE Access* (2022).
- [72] Mohammed Saeed, Nicola De Cao, and Paolo Papotti. 2023. Querying Large Language Models with SQL. *arXiv preprint arXiv:2304.00472* (2023).
- [73] Marion Schaeffer, Matthias Sesbotić, Jean-Philippe Kotowicz, Nicolas Delestre, and Cecilia Zanni-Merk. 2023. OLAF: An Ontology Learning Applied Framework. *Procedia Computer Science* 225 (2023), 2106–2115.
- [74] Priyanka Sen, Sandeep Mavadia, and Amir Saffari. 2023. Knowledge graph-augmented language models for complex question answering. (2023).
- [75] Henry Shevlin, Karina Vold, Matthew Crosby, and Marta Halina. 2019. The limits of machine intelligence. *EMBO reports* 20, 10 (2019), e49177. <https://doi.org/10.15252/embr.201949177>
- [76] Jana Straková, Eva Fučíková, Jan Hajič, and Zdeňka Urešová. 2023. Extending an Event-type Ontology: Adding Verbs and Classes Using Fine-tuned LLMs Suggestions. *arXiv:2306.02130* (2023).
- [77] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International conference on machine learning*. PMLR.
- [78] Somin Wadhwa, Silvio Amir, and Byron C. Wallace. 2023. Revisiting Relation Extraction in the era of Large Language Models. <http://arxiv.org/abs/2305.05003> arXiv:2305.05003[cs].
- [79] Zhen Wan, Fei Cheng, Zhuoyuan Mao, Qianying Liu, Haiyue Song, Jiwei Li, and Sadao Kurohashi. 2023. GPT-RE: In-context Learning for Relation Extraction using Large Language Models. arXiv:2305.02105 [cs].
- [80] 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 *Web Conference*.
- [81] Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Song. 2023. DeepStruct: Pretraining of Language Models for Structure Prediction. arXiv:2205.10475 [cs].
- [82] Liang Wang, Wei Zhao, Zhuoyu Wei, and Jingming Liu. 2022. SimKGC: Simple contrastive knowledge graph completion with pre-trained language models. *arXiv preprint arXiv:2203.02167* (2022).
- [83] Xiangyu Wang, Lyuzhou Chen, Taiyu Ban, Muhammad Usman, Yifeng Guan, Shikang Liu, Tianhao Wu, and Huanhuan Chen. 2021. Knowledge graph quality control: A survey. *Fundamental Research* 1, 5 (2021), 607–626.
- [84] Xintao Wang, Qianwen Yang, Yongting Qiu, Jiaqing Liang, Qianyi He, Zhouhong Gu, Yanghua Xiao, and Wei Wang. 2023. Knowledgept: Enhancing large language models with retrieval and storage access on knowledge bases. *arXiv preprint arXiv:2308.11761* (2023).
- [85] Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, Yong Jiang, and Wenjuan Han. 2023. Zero-Shot Information Extraction via Chatting with ChatGPT. arXiv:2302.10205 [cs].
- [86] Yanbin Wei, Qiushi Huang, Yu Zhang, and James Kwok. 2023. KICGPT: Large Language Model with Knowledge in Context for Knowledge Graph Completion. In *EMNLP*.
- [87] Xin Xie, Ningyu Zhang, Zhoubo Li, Shumin Deng, Hui Chen, Feiyu Xiong, Mosh Chen, and Huajun Chen. 2022. From discrimination to generation: Knowledge graph completion with generative transformer. In *Companion Proceedings of the Web Conference*.
- [88] Wenwen Xu, Mingzhe Fang, Li Yang, Huaxi Jiang, Geng Liang, and Chun Zuo. 2021. Enabling language representation with knowledge graph and structured semantic information. In *CCAI*. IEEE.
- [89] Xin Xu, Yuqi Zhu, Xiaohan Wang, and Ningyu Zhang. 2023. How to Unleash the Power of Large Language Models for Few-shot Relation Extraction? arXiv:2305.01555 [cs].
- [90] Linyao Yang, Hongyang Chen, Zhao Li, Xiao Ding, and Xindong Wu. 2024. Give Us the Facts: Enhancing Large Language Models with Knowledge Graphs for Fact-aware Language Modeling. arXiv:2306.11489 [cs.CL]
- [91] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. *arXiv:1809.09600* (2018).
- [92] Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. KG-BERT: BERT for knowledge graph completion. *arXiv preprint arXiv:1909.03193* (2019).
- [93] Wenhao Yu, Chenguang Zhu, Yuwei Fang, Donghan Yu, Shuohang Wang, Yichong Xu, Michael Zeng, and Meng Jiang. 2022. Dict-BERT: Enhancing Language Model Pre-training with Dictionary. In *ACL*.
- [94] Chenhan Yuan, Qianqian Xie, and Sophia Ananiadou. 2023. Zero-shot Temporal Relation Extraction with ChatGPT. <http://arxiv.org/abs/2304.05454> arXiv:2304.05454 [cs].
- [95] Amrapali Zaveri, Anisa Rula, Andrea Maurino, Ricardo Pietrobon, Jens Lehmann, and Soeren Auer. 2016. Quality assessment for linked data: A survey. *Semantic Web* 7, 1 (2016), 63–93.
- [96] Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. 2023. Universalner: Targeted distillation from large language models for open named entity recognition. *arXiv:2308.03279* (2023).