

# Unifying Large Language Models and Knowledge Graphs: A Roadmap

Shirui Pan<sup>ID</sup>, Senior Member, IEEE, Linhao Luo<sup>ID</sup>, Yufei Wang, Chen Chen<sup>ID</sup>, Jiapu Wang<sup>ID</sup>, and Xindong Wu<sup>ID</sup>, Fellow, IEEE

(Survey Paper)

**Abstract**—Large language models (LLMs), such as ChatGPT and GPT4, are making new waves in the field of natural language processing and artificial intelligence, due to their emergent ability and generalizability. However, LLMs are black-box models, which often fall short of capturing and accessing factual knowledge. In contrast, Knowledge Graphs (KGs), Wikipedia, and Huapu for example, are structured knowledge models that explicitly store rich factual knowledge. KGs can enhance LLMs by providing external knowledge for inference and interpretability. Meanwhile, KGs are difficult to construct and evolve by nature, which challenges the existing methods in KGs to generate new facts and represent unseen knowledge. Therefore, it is complementary to unify LLMs and KGs together and, simultaneously, leverage their advantages. In this article, we present a forward-looking roadmap for the unification of LLMs and KGs. Our roadmap consists of three general frameworks, namely: 1) *KG-enhanced LLMs*, which incorporate KGs during the pre-training and inference phases of LLMs, or for the purpose of enhancing understanding of the knowledge learned by LLMs; 2) *LLM-augmented KGs*, that leverage LLMs for different KG tasks such as embedding, completion, construction, graph-to-text generation, and question answering; and 3) *Synergized LLMs + KGs*, in which LLMs and KGs play equal roles and work in a mutually beneficial way to enhance both LLMs and KGs for bidirectional reasoning driven by both data and knowledge. We review and summarize existing efforts within these three frameworks in our roadmap and pinpoint their future research directions.

**Index Terms**—Natural language processing, large language models, generative pre-training, knowledge graphs, roadmap, bidirectional reasoning.

## I. INTRODUCTION

LARGE language models (LLMs)<sup>1</sup> (e.g., BERT [1], RoBERTA [2], and T5 [3]), pre-trained on the large-scale corpus, have shown great performance in various natural language processing (NLP) tasks, such as question answering [4], machine translation [5], and text generation [6]. Recently, the dramatically increasing model size further enables the LLMs with the emergent ability [7], paving the road for applying LLMs as Artificial General Intelligence (AGI). Advanced LLMs like ChatGPT<sup>2</sup> and PaLM2<sup>3</sup>, with billions of parameters, exhibit great potential in many complex practical tasks, such as education [8], code generation [9] and recommendation [10].

Despite their success in many applications, LLMs have been criticized for their lack of factual knowledge. Specifically, LLMs memorize facts and knowledge contained in the training corpus [14]. However, further studies reveal that LLMs are not able to recall facts and often experience hallucinations by generating statements that are factually incorrect [15], [28]. For example, LLMs might say “Einstein discovered gravity in 1687” when asked, “When did Einstein discover gravity?”, which contradicts the fact that Isaac Newton formulated the gravitational theory. This issue severely impairs the trustworthiness of LLMs.

As black-box models, LLMs are also criticized for their lack of interpretability. LLMs represent knowledge implicitly in their parameters. It is difficult to interpret or validate the knowledge obtained by LLMs. Moreover, LLMs perform reasoning by a probability model, which is an indecisive process [16]. The specific patterns and functions LLMs used to arrive at predictions or decisions are not directly accessible or explainable to humans [17]. Even though some LLMs are equipped to explain their predictions by applying chain-of-thought [29], their reasoning explanations also suffer from the hallucination issue [30]. This severely impairs the application of LLMs in high-stakes scenarios, such as medical diagnosis and legal judgment. For instance, in a medical diagnosis scenario, LLMs may incorrectly

Manuscript received 26 June 2023; revised 27 December 2023; accepted 5 January 2024. Date of publication 10 January 2024; date of current version 10 June 2024. This work was supported by the Australian Research Council (ARC) under Grants FT210100097 and DP240101547, and in part by the National Natural Science Foundation of China (NSFC) under Grant 62120106008. Recommended for acceptance by Yongxin Tong. (*Shirui Pan and Linhao Luo contributed equally to this work.*) (*Corresponding Author: Xindong Wu.*)

Shirui Pan is with the School of Information and Communication Technology and Institute for Integrated and Intelligent Systems (IIIS), Griffith University, Nathan, QLD 4111, Australia (e-mail: s.pan@griffith.edu.au).

Linhao Luo and Yufei Wang are with the Department of Data Science and AI, Monash University, Melbourne, VIC 3800, Australia (e-mail: linhao.luo@monash.edu; garyyufei@gmail.com).

Chen Chen is with Nanyang Technological University, Singapore 639798 (e-mail: s190009@ntu.edu.sg).

Jiapu Wang is with the Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China (e-mail: jpwang@emails.bjut.edu.cn).

Xindong Wu is with the Key Laboratory of Knowledge Engineering with Big Data (the Ministry of Education of China), Hefei University of Technology, Hefei 230002, China, and also with the Research Center for Knowledge Engineering, Zhejiang Lab, Hangzhou 310058, China (e-mail: xwu@hfut.edu.cn).

This article has supplementary downloadable material available at <https://doi.org/10.1109/TKDE.2024.3352100>, provided by the authors.

Digital Object Identifier 10.1109/TKDE.2024.3352100

<sup>1</sup>LLMs are also known as pre-trained language models (PLMs).

<sup>2</sup><https://openai.com/blog/chatgpt>

<sup>3</sup><https://ai.google/discover/palm2>

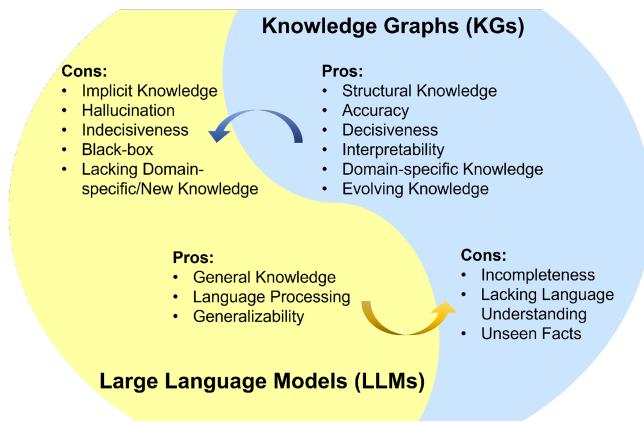


Fig. 1. Summarization of the pros and cons for LLMs and KGs. LLM pros: General Knowledge [11], Language Processing [12], Generalizability [13]; LLM cons: Implicit Knowledge [14], Hallucination [15], Indecisiveness [16], Black-box [17], Lacking Domain-specific/New Knowledge [18]. KG pros: Structural Knowledge [19], Accuracy [20], Decisiveness [21], Interpretability [22], Domain-specific Knowledge [23], Evolving Knowledge [24]; KG cons: Incompleteness [25], Lacking Language Understanding [26], Unseen Facts [27]. Pros. and Cons. are selected based on their representativeness. Detailed discussion can be found in Appendix A, available online.

diagnose a disease and provide explanations that contradict medical commonsense. This raises another issue that LLMs trained on general corpus might not be able to generalize well to specific domains or new knowledge due to the lack of domain-specific knowledge or new training data [18].

To address the above issues, a potential solution is to incorporate knowledge graphs (KGs) into LLMs. Knowledge graphs (KGs), storing enormous facts in the way of triples, i.e.,  $(\text{head entity}, \text{relation}, \text{tail entity})$ , are a structured and decisive manner of knowledge representation (e.g., Wikidata [20], YAGO [31], and NELL [32]). KGs are crucial for various applications as they offer accurate explicit knowledge [19]. Besides, they are renowned for their symbolic reasoning ability [22], which generates interpretable results. KGs can also actively evolve with new knowledge continuously added in [24]. Additionally, experts can construct domain-specific KGs to provide precise and dependable domain-specific knowledge [23].

Nevertheless, KGs are difficult to construct [25], and current approaches in KGs [27], [33], [34] are inadequate in handling the incomplete and dynamically changing nature of real-world KGs. These approaches fail to effectively model unseen entities and represent new facts. In addition, they often ignore the abundant textual information in KGs. Moreover, existing methods in KGs are often customized for specific KGs or tasks, which are not generalizable enough. Therefore, it is also necessary to utilize LLMs to address the challenges faced in KGs. We summarize the pros and cons of LLMs and KGs in Fig. 1, respectively.

Recently, the possibility of unifying LLMs with KGs has attracted increasing attention from researchers and practitioners. LLMs and KGs are inherently interconnected and can mutually enhance each other. In *KG-enhanced LLMs*, KGs can not only be incorporated into the pre-training and inference stages of LLMs to provide external knowledge [35], [36], [37], but also used for analyzing LLMs and providing interpretability [14], [38], [39].

In *LLM-augmented KGs*, LLMs have been used in various KG-related tasks, e.g., KG embedding [40], KG completion [26], KG construction [41], KG-to-text generation [42], and KGQA [43], to improve the performance and facilitate the application of KGs. In *Synergized LLM + KG*, researchers marries the merits of LLMs and KGs to mutually enhance performance in knowledge representation [44] and reasoning [45], [46]. Although there are some surveys on knowledge-enhanced LLMs [47], [48], [49], which mainly focus on using KGs as an external knowledge to enhance LLMs, they ignore other possibilities of integrating KGs for LLMs and the potential role of LLMs in KG applications.

In this article, we present a forward-looking roadmap for unifying both LLMs and KGs, to leverage their respective strengths and overcome the limitations of each approach, for various downstream tasks. We propose detailed categorization, conduct comprehensive reviews, and pinpoint emerging directions in these fast-growing fields. Our main contributions are summarized as follows:

- 1) *Roadmap*: We present a forward-looking roadmap for integrating LLMs and KGs. Our roadmap, consisting of three general frameworks to unify LLMs and KGs, namely, *KG-enhanced LLMs*, *LLM-augmented KGs*, and *Synergized LLMs + KGs*, provides guidelines for the unification of these two distinct but complementary technologies.
- 2) *Categorization and review*: For each integration framework of our roadmap, we present a detailed categorization and novel taxonomies of research on unifying LLMs and KGs. In each category, we review the research from the perspectives of different integration strategies and tasks, which provides more insights into each framework.
- 3) *Coverage of emerging advances*: We cover the advanced techniques in both LLMs and KGs. We include the discussion of state-of-the-art LLMs like ChatGPT and GPT-4 as well as the novel KGs e.g., multi-modal knowledge graphs.
- 4) *Summary of challenges and future directions*: We highlight the challenges in existing research and present several promising future research directions.

The rest of this article is organized as follows. Section II first explains the background of LLMs and KGs. Section III introduces the roadmap and the overall categorization of this article. Section IV presents the different KGs-enhanced LLM approaches. Section V describes the possible LLM-augmented KG methods. Section VI shows the approaches of synergizing LLMs and KGs. Section VII discusses the challenges and future research directions. Finally, Section VIII concludes this paper.

## II. BACKGROUND

In this section, we will first briefly introduce a few representative large language models (LLMs) and discuss the prompt engineering that efficiently uses LLMs for varieties of applications. Then, we illustrate the concept of knowledge graphs (KGs) and present different categories of KGs.

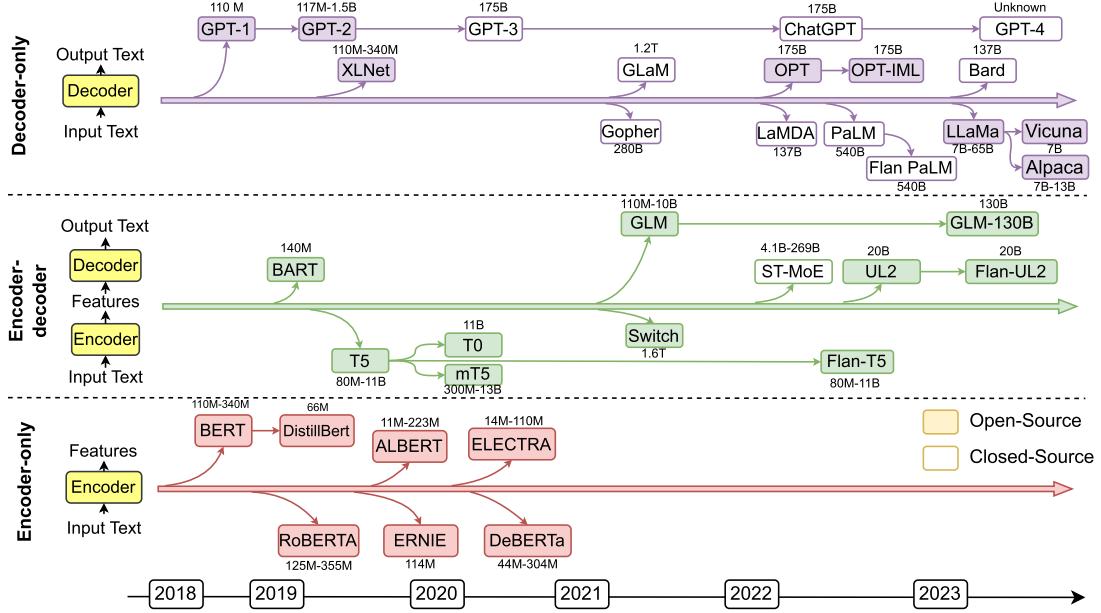


Fig. 2. Representative large language models (LLMs) in recent years. Open-source models are represented by solid squares, while closed source models are represented by hollow squares.

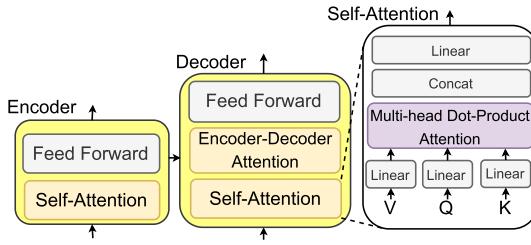


Fig. 3. Illustration of the Transformer-based LLMs with self-attention mechanism.

### A. Large Language Models (LLMs)

Large language models (LLMs) pre-trained on large-scale corpus have shown great potential in various NLP tasks [13]. As shown in Fig. 3, most LLMs derive from the Transformer design [50], which contains the encoder and decoder modules empowered by a self-attention mechanism. Based on the architecture structure, LLMs can be categorized into three groups: 1) *encoder-only LLMs*, 2) *encoder-decoder LLMs*, and 3) *decoder-only LLMs*. As shown in Fig. 2, we summarize several representative LLMs with different model architectures, model sizes, and open-source availabilities.

1) *Encoder-Only LLMs*: Encoder-only large language models only use the encoder to encode the sentence and understand the relationships between words. The common training paradigm for these model is to predict the mask words in an input sentence. This method is unsupervised and can be trained on the large-scale corpus. Encoder-only LLMs like BERT [1], ALBERT [51], RoBERTa [2], and ELECTRA [52] require adding an extra prediction head to resolve downstream tasks. These models are most effective for tasks that require understanding

the entire sentence, such as text classification [26] and named entity recognition [53].

2) *Encoder-Decoder LLMs*: Encoder-decoder large language models adopt both the encoder and decoder module. The encoder module is responsible for encoding the input sentence into a hidden-space, and the decoder is used to generate the target output text. The training strategies in encoder-decoder LLMs can be more flexible. For example, T5 [3] is pre-trained by masking and predicting spans of masking words. UL2 [54] unifies several training targets such as different masking spans and masking frequencies. Encoder-decoder LLMs (e.g., T0 [55], ST-MoE [56], and GLM-130B [57]) are able to directly resolve tasks that generate sentences based on some context, such as summarization, translation, and question answering [58].

3) *Decoder-Only LLMs*: Decoder-only large language models only adopt the decoder module to generate target output text. The training paradigm for these models is to predict the next word in the sentence. Large-scale decoder-only LLMs can generally perform downstream tasks from a few examples or simple instructions, without adding prediction heads or finetuning [59]. Many state-of-the-art LLMs (e.g., Chat-GPT [60] and GPT-4<sup>4</sup>) follow the decoder-only architecture. However, since these models are closed-source, it is challenging for academic researchers to conduct further research. Recently, Alpaca<sup>5</sup> and Vicuna<sup>6</sup> are released as open-source decoder-only LLMs. These models are finetuned based on LLaMA [61] and achieve comparable performance with ChatGPT and GPT-4.

4) *Prompt Engineering*: Prompt engineering is a novel field that focuses on creating and refining prompts to maximize the

<sup>4</sup><https://openai.com/product/gpt-4>

<sup>5</sup>[https://github.com/tatsu-lab/stanford\\_alpaca](https://github.com/tatsu-lab/stanford_alpaca)

<sup>6</sup><https://lmsys.org/blog/2023-03-30-vicuna/>

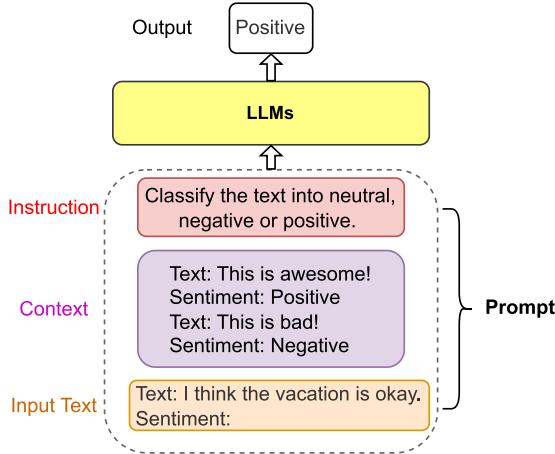


Fig. 4. Example of sentiment classification prompt.

effectiveness of large language models (LLMs) across various applications and research areas [62]. As shown in Fig. 4, a prompt is a sequence of natural language inputs for LLMs that are specified for the task, such as sentiment classification. A prompt could contain several elements, i.e., 1) *Instruction*, 2) *Context*, and 3) *Input Text*. *Instruction* is a short sentence that instructs the model to perform a specific task. *Context* provides the context for the input text or few-shot examples. *Input Text* is the text that needs to be processed by the model.

Prompt engineering seeks to improve the capacity of large language models (e.g., ChatGPT) in diverse complex tasks such as question answering, sentiment classification, and common sense reasoning. Chain-of-thought (CoT) prompt [63] enables complex reasoning capabilities through intermediate reasoning steps. Prompt engineering also enables the integration of structural data like knowledge graphs (KGs) into LLMs. Li et al. [64] simply linearizes the KGs and uses templates to convert the KGs into passages. Mindmap [65] designs a KG prompt to convert graph structure into a mind map that enables LLMs to perform reasoning on it. Prompt offers a simple way to utilize the potential of LLMs without finetuning. Proficiency in prompt engineering leads to a better understanding of the strengths and weaknesses of LLMs.

## B. Knowledge Graphs (KGs)

Knowledge graphs (KGs) store structured knowledge as a collection of triples  $\mathcal{KG} = \{(h, r, t) \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}\}$ , where  $\mathcal{E}$  and  $\mathcal{R}$  respectively denote the set of entities and relations. Existing knowledge graphs (KGs) can be classified into four groups based on the stored information: 1) *encyclopedic KGs*, 2) *commonsense KGs*, 3) *domain-specific KGs*, and 4) *multi-modal KGs*. We illustrate the examples of KGs of different categories in Fig. 5.

1) *Encyclopedic Knowledge Graphs*: Encyclopedic knowledge graphs are the most ubiquitous KGs, which represent the general knowledge in real-world. Encyclopedic knowledge graphs are often constructed by integrating information from diverse and extensive sources, including human experts, encyclopedias, and databases. Wikidata [20] is one of the most widely

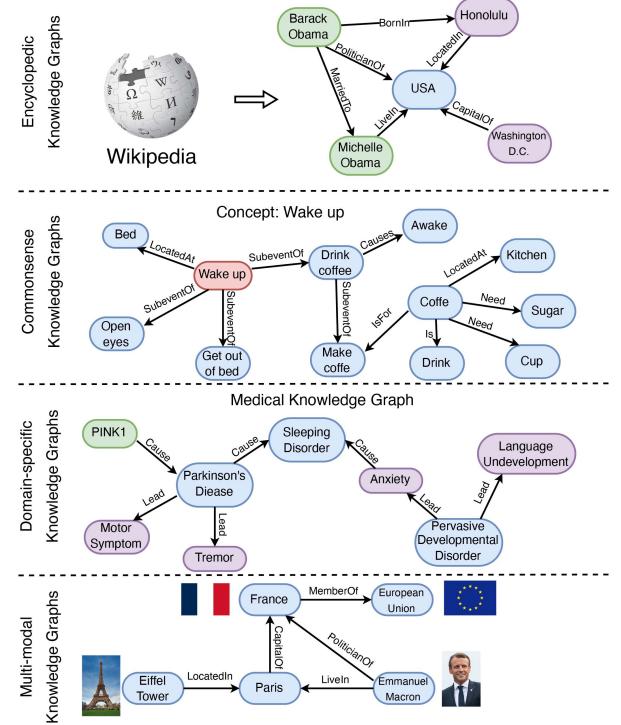


Fig. 5. Examples of different categories' knowledge graphs, i.e., *encyclopedic KGs*, *commonsense KGs*, *domain-specific KGs*, and *multi-modal KGs*.

used encyclopedic knowledge graphs, which incorporates varieties of knowledge extracted from articles on Wikipedia. Other typical encyclopedic knowledge graphs, like Freebase [66], Dbpedia [67], and YAGO [31] are also derived from Wikipedia. In addition, NELL [32] is a continuously improving encyclopedic knowledge graph, which automatically extracts knowledge from the web, and uses that knowledge to improve its performance over time. There are several encyclopedic knowledge graphs available in languages other than English such as CN-DBpedia [68] and Vikidia [69]. The largest knowledge graph, named Knowledge Ocean (KO)<sup>7</sup>, currently contains 4,8784,3636 entities and 17,3115,8349 relations in both English and Chinese.

2) *Commonsense Knowledge Graphs*: Commonsense knowledge graphs formulate the knowledge about daily concepts, e.g., objects, and events, as well as their relationships [70]. Compared with encyclopedic knowledge graphs, commonsense knowledge graphs often model the tacit knowledge extracted from text such as (*Car*, *UsedFor*, *Drive*). ConceptNet [71] contains a wide range of commonsense concepts and relations, which can help computers understand the meanings of words people use. ATOMIC [72], [73] and ASER [74] focus on the causal effects between events, which can be used for commonsense reasoning. Some other commonsense knowledge graphs, such as TransOMCS [75] and CausalBank [76] are automatically constructed to provide commonsense knowledge.

<sup>7</sup><https://ko.zhonghuapu.com/>

TABLE I  
REPRESENTATIVE APPLICATIONS OF USING LLMs AND KGs

Name	Category	LLMs	KGs	URL
ChatGPT/GPT-4	Chat Bot	✓		<a href="https://shorturl.at/cmsE0">https://shorturl.at/cmsE0</a>
ERNIE 3.0	Chat Bot	✓	✓	<a href="https://shorturl.at/sCLV9">https://shorturl.at/sCLV9</a>
Bard	Chat Bot	✓	✓	<a href="https://shorturl.at/pDLV6">https://shorturl.at/pDLV6</a>
Firefly	Photo Editing	✓		<a href="https://shorturl.at/fkzJv">https://shorturl.at/fkzJv</a>
AutoGPT	AI Assistant	✓		<a href="https://shorturl.at/bkoSY">https://shorturl.at/bkoSY</a>
Copilot	Coding Assistant	✓		<a href="https://shorturl.at/IKLUV">https://shorturl.at/IKLUV</a>
New Bing	Web Search	✓		<a href="https://shorturl.at/bimps">https://shorturl.at/bimps</a>
Shop.ai	Recommendation	✓		<a href="https://shorturl.at/alcY5">https://shorturl.at/alcY5</a>
Wikidata	Knowledge Base		✓	<a href="https://shorturl.at/lyMY5">https://shorturl.at/lyMY5</a>
KO	Knowledge Base		✓	<a href="https://shorturl.at/sx238">https://shorturl.at/sx238</a>
OpenBG	Recommendation		✓	<a href="https://shorturl.at/pDMV9">https://shorturl.at/pDMV9</a>
Doctor.ai	Health Care Assistant	✓	✓	<a href="https://shorturl.at/dhlK0">https://shorturl.at/dhlK0</a>

3) *Domain-Specific Knowledge Graphs*: Domain-specific knowledge graphs are often constructed to represent knowledge in a specific domain, e.g., medical, biology, and finance [23]. Compared with encyclopedic knowledge graphs, domain-specific knowledge graphs are often smaller in size, but more accurate and reliable. For example, UMLS [77] is a domain-specific knowledge graph in the medical domain, which contains biomedical concepts and their relationships. In addition, there are some domain-specific knowledge graphs in other domains, such as finance [78], geology [79], biology [80], chemistry [81] and genealogy [82].

4) *Multi-Modal Knowledge Graphs*: Unlike conventional knowledge graphs that only contain textual information, multi-modal knowledge graphs represent facts in multiple modalities such as images, sounds, and videos [83]. For example, IMGpedia [84], MMKG [85], and Richpedia [86] incorporate both the text and image information into the knowledge graphs. These knowledge graphs can be used for various multi-modal tasks such as image-text matching [87], visual question answering [88], and recommendation [89].

### C. Applications

LLMs as KGs have been widely applied in various real-world applications. We summarize some representative applications of using LLMs and KGs in Table I. ChatGPT/GPT-4 are LLM-based chatbots that can communicate with humans in a natural dialogue format. To improve knowledge awareness of LLMs, ERNIE 3.0 and Bard incorporate KGs into their chatbot applications. Instead of Chatbot, Firefly develops a photo editing application that allows users to edit photos by using natural language descriptions. Copilot, New Bing, and Shop.ai adopt LLMs to empower their applications in the areas of coding assistant, web search, and recommendation, respectively. Wikidata and KO are two representative knowledge graph applications that are used to provide external knowledge. OpenBG [90] is a knowledge graph designed for recommendation. Doctor.ai develops a health care assistant that incorporates LLMs and KGs to provide medical advice.

## III. ROADMAP & CATEGORIZATION

In this section, we first present a road map of explicit frameworks that unify LLMs and KGs. Then, we present the categorization of research on unifying LLMs and KGs.

### A. Roadmap

The roadmap of unifying KGs and LLMs is illustrated in Fig. 6. In the roadmap, we identify three frameworks for the unification of LLMs and KGs, including KG-enhanced LLMs, LLM-augmented KGs, and Synergized LLMs + KGs. The KG-enhanced LLMs and LLM-augmented KGs are two parallel frameworks that aim to enhance the capabilities of LLMs and KGs, respectively. Building upon these frameworks, Synergized LLMs + KGs is a unified framework that aims to synergize LLMs and KGs to mutually enhance each other.

1) *Kg-Enhanced LLMs*: LLMs are renowned for their ability to learn knowledge from large-scale corpus and achieve state-of-the-art performance in various NLP tasks. However, LLMs are often criticized for their hallucination issues [15], and lacking of interpretability. To address these issues, researchers have proposed to enhance LLMs with knowledge graphs (KGs).

KGs store enormous knowledge in an explicit and structured way, which can be used to enhance the knowledge awareness of LLMs. Some researchers have proposed to incorporate KGs into LLMs during the pre-training stage, which can help LLMs learn knowledge from KGs [35], [91]. Other researchers have proposed to incorporate KGs into LLMs during the inference stage. By retrieving knowledge from KGs, it can significantly improve the performance of LLMs in accessing domain-specific knowledge [92]. To improve the interpretability of LLMs, researchers also utilize KGs to interpret the facts [14] and the reasoning process of LLMs [38].

2) *Llm-Augmented KGs*: KGs store structure knowledge playing an essential role in many real-word applications [19]. Existing methods in KGs fall short of handling incomplete KGs [33] and processing text corpus to construct KGs [93]. With the generalizability of LLMs, many researchers are trying to harness the power of LLMs for addressing KG-related tasks.

The most straightforward way to apply LLMs as text encoders for KG-related tasks. Researchers take advantage of LLMs to process the textual corpus in the KGs and then use the representations of the text to enrich KGs representation [94]. Some studies also use LLMs to process the original corpus and extract relations and entities for KG construction [95]. Recent studies try to design a KG prompt that can effectively convert structural KGs into a format that can be comprehended by LLMs. In this way, LLMs can be directly applied to KG-related tasks, e.g., KG completion [96] and KG reasoning [97].

3) *Synergized LLMs + KGs*: The synergy of LLMs and KGs has attracted increasing attention from researchers these years [40], [42]. LLMs and KGs are two inherently complementary techniques, which should be unified into a general framework to mutually enhance each other.

To further explore the unification, we propose a unified framework of the synergized LLMs + KGs in Fig. 7. The unified framework contains four layers: 1) *Data*, 2) *Synergized Model*, 3) *Technique*, and 4) *Application*. In the *Data* layer, LLMs and KGs are used to process the textual and structural data, respectively. With the development of multi-modal LLMs [98] and KGs [99], this framework can be extended to process multi-modal data, such as video, audio, and images. In the *Synergized Model* layer, LLMs and KGs could synergize with each other to

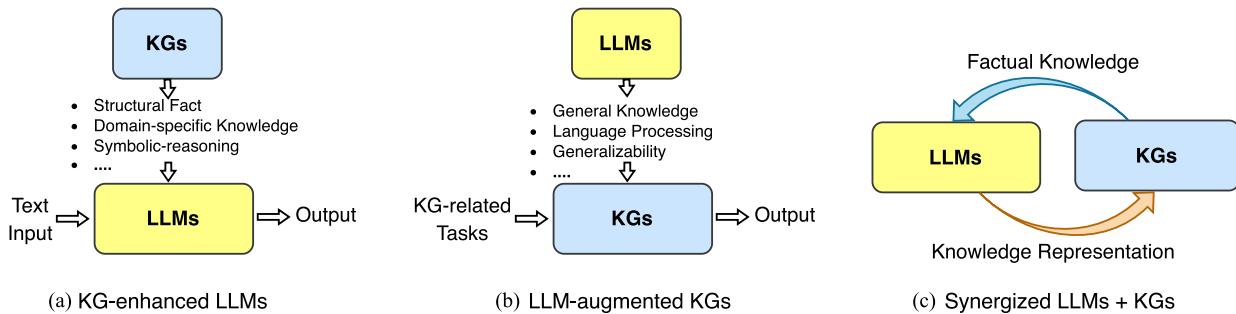


Fig. 6. General roadmap of unifying KGs and LLMs.

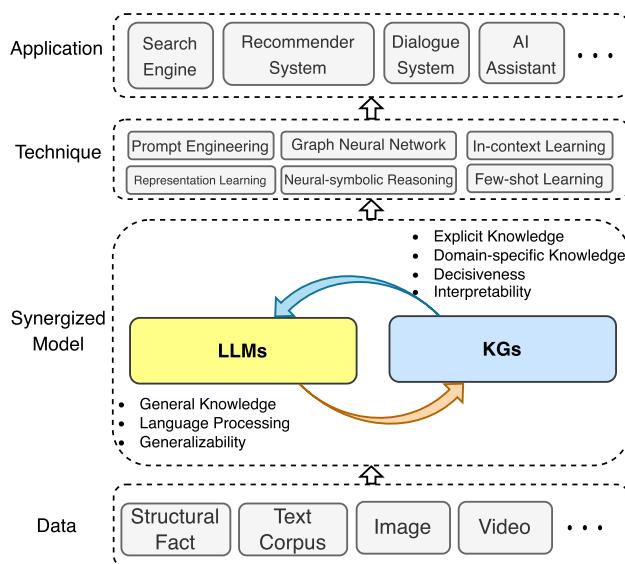


Fig. 7. General framework of the Synergized LLMs + KGs, which contains four layers: 1) Data, 2) Synergized Model, 3) Technique, and 4) Application.

improve their capabilities. In *Technique* layer, related techniques that have been used in LLMs and KGs can be incorporated into this framework to further enhance the performance. In the *Application* layer, LLMs and KGs can be integrated to address various real-world applications, such as search engines [100], recommender systems [10], and AI assistants [101].

### B. Categorization

To better understand the research on unifying LLMs and KGs, we further provide a fine-grained categorization for each framework in the roadmap. Specifically, we focus on different ways of integrating KGs and LLMs, i.e., KG-enhanced LLMs, KG-augmented LLMs, and Synergized LLMs + KGs. The fine-grained categorization of the research is illustrated in Fig. 8.

**KG-enhanced LLMs:** Integrating KGs can enhance the performance and interpretability of LLMs in various downstream tasks. We categorize the research on KG-enhanced LLMs into three groups:

- 1) *KG-enhanced LLM pre-training* includes works that apply KGs during the pre-training stage and improve the knowledge expression of LLMs.
  - 2) *KG-enhanced LLM inference* includes research that utilizes KGs during the inference stage of LLMs, which enables LLMs to access the latest knowledge without retraining.
  - 3) *KG-enhanced LLM interpretability* includes works that use KGs to understand the knowledge learned by LLMs and interpret the reasoning process of LLMs.

*LLM-augmented KGs:* LLMs can be applied to augment various KG-related tasks. We categorize the research on LLM-augmented KGs into five groups based on the task types:

- 1) *LLM-augmented KG embedding* includes studies that apply LLMs to enrich representations of KGs by encoding the textual descriptions of entities and relations.
  - 2) *LLM-augmented KG completion* includes papers that utilize LLMs to encode text or generate facts for better KGC performance.
  - 3) *LLM-augmented KG construction* includes works that apply LLMs to address the entity discovery, coreference resolution, and relation extraction tasks for KG construction.
  - 4) *LLM-augmented KG-to-text Generation* includes research that utilizes LLMs to generate natural language that describes the facts from KGs.
  - 5) *LLM-augmented KG question answering* includes studies that apply LLMs to bridge the gap between natural language questions and retrieve answers from KGs.

*Synergized LLMs + KGs:* The synergy of LLMs and KGs aims to integrate LLMs and KGs into a unified framework to mutually enhance each other. In this categorization, we review the recent attempts of Synergized LLMs + KGs from the perspectives of *knowledge representation* and *reasoning*.

In the following sections (Sec Sections IV, V, and VI), we will provide details on these categorizations.

#### IV. KG-ENHANCED LLMS

Large language models (LLMs) achieve promising results in many natural language processing tasks. However, LLMs have been criticized for their lack of practical knowledge and tendency to generate factual errors during inference. To address

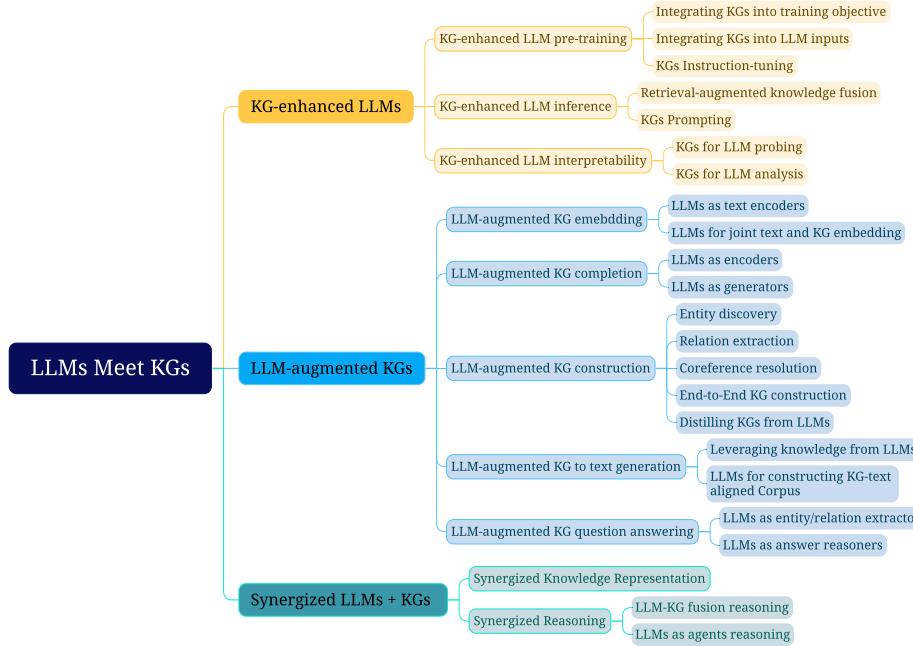


Fig. 8. Fine-grained categorization of research on unifying large language models (LLMs) with knowledge graphs (KGs).

TABLE II  
SUMMARY OF KG-ENHANCED LLM METHODS

Task	Method	Year	KG	Technique
KG-enhanced LLM pre-training	ERNIE [35]	2019	E	Integrating KGs into Training Objective
	GLM [102]	2019	C	Integrating KGs into Training Objective
	E-BERT [103]	2020	D	Integrating KGs into Training Objective
	KEPLER [40]	2021	E	Integrating KGs into Training Objective
	Deterministic LLM [104]	2022	E	Integrating KGs into Training Objective
	KALA [105]	2022	D	Integrating KGs into Training Objective
	WALK [106]	2022	E	Integrating KGs into Training Objective
	K-PALM [109]	2022	E + D	Integrating KGs into Language Model Inputs
	GLoAKME [107]	2022	E + D	Integrating KGs into Language Model Inputs
	ERNIE3.0 [101]	2021	E + D	Integrating KGs into Language Model Inputs
	DALLM [108]	2022	E	Integrating KGs into Language Model Inputs
	KIP-PLM [109]	2022	E	KGs Instruction-tuning
	OntoPrompt [110]	2022	E + D	KGs Instruction-tuning
	ChatKBQA [111]	2023	E	KGs Instruction-tuning
	RoG [112]	2023	E	KGs Instruction-tuning
KG-enhanced LLM inference	KGlm [113]	2019	E	Retrieval-augmented knowledge fusion
	RAG [92]	2020	E	Retrieval-augmented knowledge fusion
	EMAT [115]	2022	E	Retrieval-augmented knowledge fusion
	Li et al. [164]	2023	C	KGs Prompting
	Mindmap [65]	2023	E + D	KGs Prompting
	ChatRule [116]	2023	E + D	KGs Prompting
	CoK [117]	2023	E + C + D	KGs Prompting
KG-enhanced LLM interpretability	LAMA [14]	2019	E	KGs for LLM probing
	LM2BERT [118]	2020	E	KGs for LLM probing
	Autoprompt [119]	2020	E	KGs for LLM probing
	MedLAMA [120]	2022	D	KGs for LLM probing
	LLM-factevel [121]	2023	E + D	KGs for LLM probing
	KapNet [38]	2019	C	KGs for LLM analysis
	Interpret-Im [122]	2021	E	KGs for LLM analysis
	knowledge-neurons [39]	2021	E	KGs for LLM analysis
	Shaobo et al. [123]	2022	E	KGs for LLM analysis

E: Encyclopedic Knowledge Graphs, C: Commonsense Knowledge Graphs, D: Domain-Specific Knowledge Graphs.

this issue, researchers have proposed integrating knowledge graphs (KGs) to enhance LLMs. In this section, we first introduce the KG-enhanced LLM pre-training, which aims to inject knowledge into LLMs during the pre-training stage. Then, we introduce the KG-enhanced LLM inference, which enables LLMs to consider the latest knowledge while generating sentences. Finally, we introduce the KG-enhanced LLM interpretability, which aims to improve the interpretability of LLMs by using KGs. Table II summarizes the typical methods that integrate KGs for LLMs.

#### A. KG-Enhanced LLM Pre-Training

Existing large language models mostly rely on unsupervised training on the large-scale corpus. While these models may exhibit impressive performance on downstream tasks, they often lack practical knowledge relevant to the real world. Previous works that integrate KGs into large language models can be categorized into three parts: 1) *Integrating KGs into training*

*objective*, 2) *Integrating KGs into LLM inputs*, and 3) *KGs Instruction-tuning*.

1) *Integrating KGs Into Training Objective*: The research efforts in this category focus on designing novel knowledge-aware training objectives. An intuitive idea is to expose more knowledge entities in the pre-training objective. GLM [102] leverages the knowledge graph structure to assign a masking probability. Specifically, entities that can be reached within a certain number of hops are considered to be the most important entities for learning, and they are given a higher masking probability during pre-training. Furthermore, E-BERT [103] further controls the balance between the token-level and entity-level training losses. The training loss values are used as indications of the learning process for token and entity, which dynamically determines their ratio for the next training epochs. SKEP [124] also follows a similar fusion to inject sentiment knowledge during LLMs pre-training. SKEP first determines words with positive and negative sentiment by utilizing PMI along with a predefined set of seed sentiment words. Then, it assigns a higher masking probability to those identified sentiment words in the word masking objective.

The other line of work explicitly leverages the connections with knowledge and input text. As shown in Fig. 9, ERNIE [35] proposes a novel word-entity alignment training objective as a pre-training objective. Specifically, ERNIE feeds both sentences and corresponding entities mentioned in the text into LLMs, and then trains the LLMs to predict alignment links between textual tokens and entities in knowledge graphs. Similarly, KALM [91] enhances the input tokens by incorporating entity embeddings and includes an entity prediction pre-training task in addition to the token-only pre-training objective. This approach aims to improve the ability of LLMs to capture knowledge related to entities. Finally, KEPLER [40] directly employs both knowledge graph embedding training objective and Masked token pre-training objective into a shared transformer-based encoder.

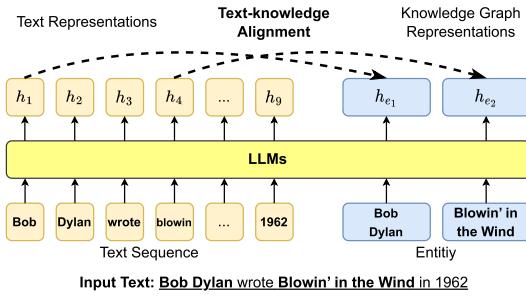


Fig. 9. Injecting KG information into LLMs training objective via text-knowledge alignment loss, where  $h$  denotes the hidden representation generated by LLMs.

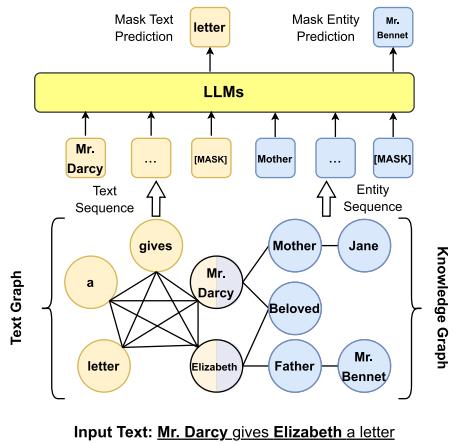


Fig. 10. Injecting KG information into LLMs inputs using graph structure.

Deterministic LLM [104] focuses on pre-training language models to capture *deterministic* factual knowledge. It only masks the span that has a deterministic entity as the question and introduces additional clue contrast learning and clue classification objective. WKLM [106] first replaces entities in the text with other same-type entities and then feeds them into LLMs. The model is further pre-trained to distinguish whether the entities have been replaced or not.

2) *Integrating KGs Into LLM Inputs:* As shown in Fig. 10, this kind of research focus on introducing relevant knowledge sub-graph into the inputs of LLMs. Given a knowledge graph triple and the corresponding sentences, ERNIE 3.0 [101] represents the triple as a sequence of tokens and directly concatenates them with the sentences. It further randomly masks either the relation token in the triple or tokens in the sentences to better combine knowledge with textual representations. However, such direct knowledge triple concatenation method allows the tokens in the sentence to intensively interact with the tokens in the knowledge sub-graph, which could result in *Knowledge Noise* [36]. To solve this issue, K-BERT [36] takes the first step to inject the knowledge triple into the sentence via a *visible matrix* where only the knowledge entities have access to the knowledge triple information, while the tokens in the sentences can only see each other in the self-attention module. To further reduce *Knowledge Noise*, Colake [107] proposes a unified word-knowledge graph (shown in Fig. 10) where the tokens in

the input sentences form a fully connected word graph where tokens aligned with knowledge entities are connected with their neighboring entities.

The above methods can indeed inject a large amount of knowledge into LLMs. However, they mostly focus on popular entities and overlook the low-frequent and long-tail ones. DkLLM [108] aims to improve the LLMs representations towards those entities. DkLLM first proposes a novel measurement to determine long-tail entities and then replaces these selected entities in the text with pseudo token embedding as new input to the large language models. Furthermore, Dict-BERT [125] proposes to leverage external dictionaries to solve this issue. Specifically, Dict-BERT improves the representation quality of rare words by appending their definitions from the dictionary at the end of input text and trains the language model to locally align rare word representations in input sentences and dictionary definitions as well as to discriminate whether the input text and definition are correctly mapped.

3) *KGs Instruction-Tuning:* Instead of injecting factual knowledge into LLMs, the KGs Instruction-tuning aims to finetune LLMs to better comprehend the structure of KGs and effectively follow user instructions to conduct complex tasks. KGs Instruction-tuning utilizes both facts and the structure of KGs to create instruction-tuning datasets. LLMs finetuned on these datasets can extract both factual and structural knowledge from KGs, enhancing the reasoning ability of LLMs. KP-PLM [109] first designs several prompt templates to transfer structural graphs into natural language text. Then, two self-supervised tasks are proposed to finetune LLMs to further leverage the knowledge from these prompts. OntoPrompt [110] proposes an ontology-enhanced prompt-tuning that can place knowledge of entities into the context of LLMs, which are further finetuned on several downstream tasks. ChatKBQA [111] finetunes LLMs on KG structure to generate logical queries, which can be executed on KGs to obtain answers. To better reason on graphs, RoG [112] presents a planning-retrieval-reasoning framework. RoG is finetuned on KG structure to generate relation paths grounded by KGs as faithful plans. These plans are then used to retrieve valid reasoning paths from the KGs for LLMs to conduct faithful reasoning and generate interpretable results.

KGs Instruction-tuning can better leverage the knowledge from KGs for downstream tasks. However, it requires retraining the models, which is time-consuming and requires lots of resources.

## B. KG-Enhanced LLM Inference

The above methods could effectively fuse knowledge into LLMs. However, real-world knowledge is subject to change and the limitation of these approaches is that they do not permit updates to the incorporated knowledge without retraining the model. As a result, they may not generalize well to the unseen knowledge during inference [126]. Therefore, considerable research has been devoted to keeping the knowledge space and text space separate and injecting the knowledge while inference. These methods mostly focus on the Question Answering (QA) tasks, because QA requires the model to capture both textual semantic meanings and up-to-date real-world knowledge.

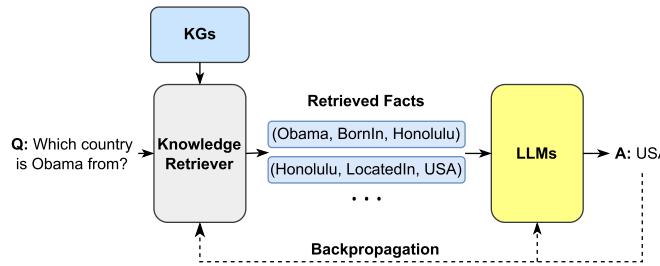


Fig. 11. Retrieving external knowledge to enhance the LLM generation.

*1) Retrieval-Augmented Knowledge Fusion:* Retrieval-Augmented Knowledge Fusion is a popular method to inject knowledge into LLMs during inference. The key idea is to retrieve relevant knowledge from a large corpus and then fuse the retrieved knowledge into LLMs. As shown in Fig. 11, RAG [92] proposes to combine non-parametric and parametric modules to handle the external knowledge. Given the input text, RAG first searches for relevant KG in the non-parametric module via MIPS to obtain several documents. RAG then treats these documents as hidden variables  $z$  and feeds them into the output generator, empowered by Seq2Seq LLMs, as additional context information. The research indicates that using different retrieved documents as conditions at different generation steps performs better than only using a single document to guide the whole generation process. The experimental results show that RAG outperforms other parametric-only and non-parametric-only baseline models in open-domain QA. RAG can also generate more specific, diverse, and factual text than other parameter-only baselines. Story-fragments [127] further improves architecture by adding an additional module to determine salient knowledge entities and fuse them into the generator to improve the quality of generated long stories. EMAT [115] further improves the efficiency of such a system by encoding external knowledge into a key-value memory and exploiting the fast maximum inner product search for memory querying. REALM [114] proposes a novel knowledge retriever to help the model to retrieve and attend over documents from a large corpus during the pre-training stage and successfully improves the performance of open-domain question answering. KGLM [113] selects the facts from a knowledge graph using the current context to generate factual sentences. With the help of an external knowledge graph, KGLM could describe facts using out-of-domain words or phrases.

*2) KGs Prompting:* To better feed the KG structure into the LLM during inference, KGs prompting aims to design a crafted prompt that converts structured KGs into text sequences, which can be fed as context into LLMs. In this way, LLMs can better take advantage of the structure of KGs to perform reasoning. Li et al. [64] adopt the pre-defined template to convert each triple into a short sentence, which can be understood by LLMs for reasoning. Mindmap [65] designs a KG prompt to convert graph structure into a mind map that enables LLMs to perform reasoning by consolidating the facts in KGs and the implicit knowledge from LLMs. ChatRule [116] samples several relation paths from KGs, which are verbalized and fed into LLMs. Then, LLMs

are prompted to generate meaningful logical rules that can be used for reasoning. CoK [117] proposes a chain-of-knowledge prompting that uses a sequence of triples to elicit the reasoning ability of LLMs to reach the final answer.

KGs prompting presents a simple way to synergize LLMs and KGs. By using the prompt, we can easily harness the power of LLMs to perform reasoning based on KGs without retraining the models. However, the prompt is usually designed manually, which requires lots of human effort.

### C. Comparison Between KG-Enhanced LLM Pre-Training and Inference

KG-enhanced LLM Pre-training methods commonly enrich large-amount of unlabeled corpus with semantically relevant real-world knowledge. These methods allow the knowledge representations to be aligned with appropriate linguistic context and explicitly train LLMs to leverage those knowledge from scratch. When applying the resulting LLMs to downstream knowledge-intensive tasks, they should achieve optimal performance. In contrast, KG-enhanced LLM inference methods only present the knowledge to LLMs in the inference stage and the underlying LLMs may not be trained to fully leverage these knowledge when conducting downstream tasks, potentially resulting in sub-optimal model performance.

However, real-world knowledge is dynamic and requires frequent updates. Despite being effective, the KG-enhanced LLM Pre-training methods never permit knowledge updates or editing without model re-training. As a result, the KG-enhanced LLM Pre-training methods could generalize poorly to recent or unseen knowledge. KG-enhanced LLM inference methods can easily maintain knowledge updates by changing the inference inputs. These methods help improve LLMs performance on new knowledge and domains.

In summary, when to use these methods depends on the application scenarios. If one wishes to apply LLMs to handle time-insensitive knowledge in particular domains (e.g., commonsense and reasoning knowledge), KG-enhanced LLM Pre-training methods should be considered. Otherwise, KG-enhanced LLM inference methods can be used to handle open-domain knowledge with frequent updates.

### D. KG-Enhanced LLM Interpretability

Although LLMs have achieved remarkable success in many NLP tasks, they are still criticized for their lack of interpretability. The large language model (LLM) interpretability refers to the understanding and explanation of the inner workings and decision-making processes of a large language model [17]. This can improve the trustworthiness of LLMs and facilitate their applications in high-stakes scenarios such as medical diagnosis and legal judgment. Knowledge graphs (KGs) represent the knowledge structurally and can provide good interpretability for the reasoning results. Therefore, researchers try to utilize KGs to improve the interpretability of LLMs, which can be roughly grouped into two categories: 1) *KGs for language model probing*, and 2) *KGs for language model analysis*.

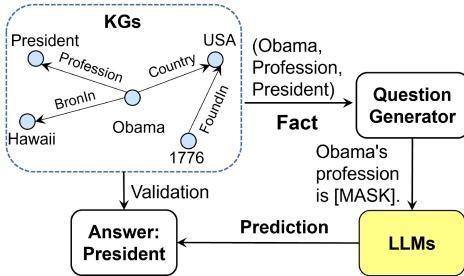


Fig. 12. General framework of using knowledge graph for language model probing.

**1) KGs for LLM Probing:** The large language model (LLM) probing aims to understand the knowledge stored in LLMs. LLMs, trained on large-scale corpus, are often known as containing enormous knowledge. However, LLMs store the knowledge in a hidden way, making it hard to figure out the stored knowledge. Moreover, LLMs suffer from the hallucination problem [15], which results in generating statements that contradict facts. This issue significantly affects the reliability of LLMs. Therefore, it is necessary to probe and verify the knowledge stored in LLMs.

LAMA [14] is the first work to probe the knowledge in LLMs by using KGs. As shown in Fig. 12, LAMA first converts the facts in KGs into cloze statements by a pre-defined prompt template and then uses LLMs to predict the missing entity. The prediction results are used to evaluate the knowledge stored in LLMs. For example, we try to probe whether LLMs know the fact (*Obama, profession, president*). We first convert the fact triple into a cloze question “Obama’s profession is \_.” with the object masked. Then, we test if the LLMs can predict the object “president” correctly.

However, LAMA ignores the fact that the prompts are inappropriate. For example, the prompt “*Obama worked as a \_*” may be more favorable to the prediction of the blank by the language models than “*Obama is a \_ by profession*”. Thus, LPAQA [118] proposes a mining and paraphrasing-based method to automatically generate high-quality and diverse prompts for a more accurate assessment of the knowledge contained in the language model. Moreover, Adolphs et al. [128] attempt to use examples to make the language model understand the query, and experiments obtain substantial improvements for BERT-large on the T-REx data. Unlike using manually defined prompt templates, Autoprompt [119] proposes an automated method, which is based on the gradient-guided search to create prompts. LLM-factevel [121] designs a systematic framework that automatically generates probing questions from KGs. The generated questions are then used to evaluate the factual knowledge stored in LLMs.

Instead of probing the general knowledge by using the encyclopedic and commonsense knowledge graphs, BioLAMA [129] and MedLAMA [120] probe the medical knowledge in LLMs by using medical knowledge graphs. Alex et al. [130] investigate the capacity of LLMs to retain less popular factual knowledge. They select unpopular facts from Wikidata knowledge graphs which have low-frequency clicked entities. These facts are then

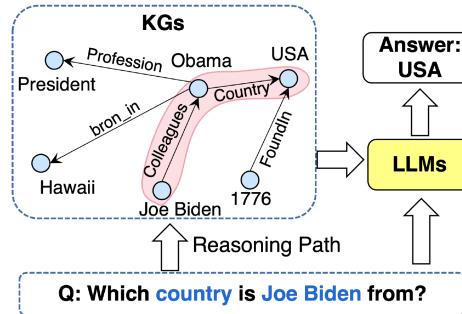


Fig. 13. General framework of using knowledge graph for language model analysis.

used for the evaluation, where the results indicate that LLMs encounter difficulties with such knowledge, and that scaling fails to appreciably improve memorization of factual knowledge in the tail.

**2) KGs for LLM Analysis:** Knowledge graphs (KGs) for pre-train language models (LLMs) analysis aims to answer the following questions such as “how do LLMs generate the results?”, and “how do the function and structure work in LLMs?”. To analyze the inference process of LLMs, as shown in Fig. 13, KagNet [38] and QA-GNN [131] make the results generated by LLMs at each reasoning step grounded by knowledge graphs. In this way, the reasoning process of LLMs can be explained by extracting the graph structure from KGs. Shaobo et al. [123] investigate how LLMs generate the results correctly. They adopt the causal-inspired analysis from facts extracted from KGs. This analysis quantitatively measures the word patterns that LLMs depend on to generate the results. The results show that LLMs generate the missing factual more by the positionally closed words rather than the knowledge-dependent words. Thus, they claim that LLMs are inadequate to memorize factual knowledge because of the inaccurate dependence. To interpret the training of LLMs, Swamy et al. [122] adopt the language model during pre-training to generate knowledge graphs. The knowledge acquired by LLMs during training can be unveiled by the facts in KGs explicitly. To explore how implicit knowledge is stored in parameters of LLMs, Dai et al. [39] propose the concept of *knowledge neurons*. Specifically, activation of the identified knowledge neurons is highly correlated with knowledge expression. Thus, they explore the knowledge and facts represented by each neuron by suppressing and amplifying knowledge neurons.

## V. LLM-AUGMENTED KGs

Knowledge graphs are famous for representing knowledge in a structural manner. They have been applied in many downstream tasks such as question answering, recommendation, and web search. However, the conventional KGs are often incomplete and existing methods often lack considering textual information. To address these issues, recent research has explored integrating LLMs to augment KGs to consider the textual information and improve the performance in downstream tasks. In this section, we will introduce the recent research on LLM-augmented KGs. We will introduce the methods that

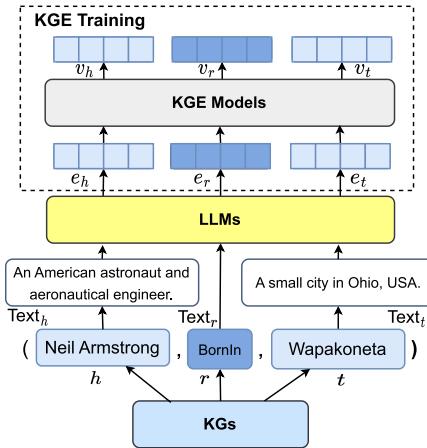


Fig. 14. LLMs as text encoder for knowledge graph embedding (KGE).

integrate LLMs for KG embedding, KG completion, KG construction, KG-to-text generation, and KG question answering, respectively. Table that summarizes representative works can be found in the Appendix B, available online.

#### A. LLM-Augmented KG Embedding

Knowledge graph embedding (KGE) aims to map each entity and relation into a low-dimensional vector (embedding) space. These embeddings contain both semantic and structural information of KGs, which can be utilized for various tasks such as question answering [132], reasoning [38], and recommendation [133].

**LLMs as Text Encoders:** Conventional knowledge graph embedding methods mainly rely on the structural information of KGs to optimize a scoring function defined on embeddings (e.g., TransE [33], and DisMult [134]). However, these approaches often fall short in representing unseen entities and long-tailed relations due to their limited structural connectivity [135], [136]. To address this issue, as shown in Fig. 14, recent research adopt LLMs to enrich representations of KGs by encoding the textual descriptions of entities and relations [40], [94].

**LLMs for Joint Text and KG Embedding:** Instead of using KGE model to consider graph structure, another line of methods directly employs LLMs to incorporate both the graph structure and textual information into the embedding space simultaneously [137], [138], [139]. As shown in Fig. 15, they treat the entities and relations as special tokens in the LLM. During training, it transfers each triple and corresponding text description into a sentence where the tailed entities are replaced by [MASK]. The sentence is fed into a LLM, which then finetunes the model to predict the masked entity.

More details about LLM-augmented KG embedding can be found in the Appendix B.1, available online.

#### B. LLM-Augmented KG Completion

Knowledge Graph Completion (KGC) refers to the task of inferring missing facts in a given knowledge graph. Similar

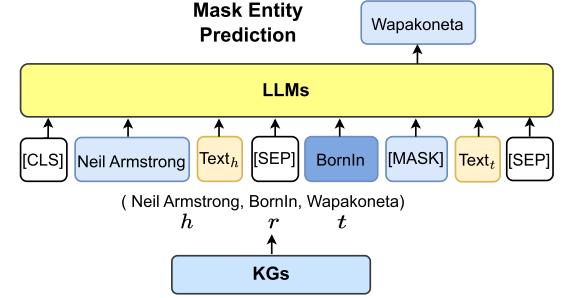


Fig. 15. LLMs for joint text and knowledge graph embedding.

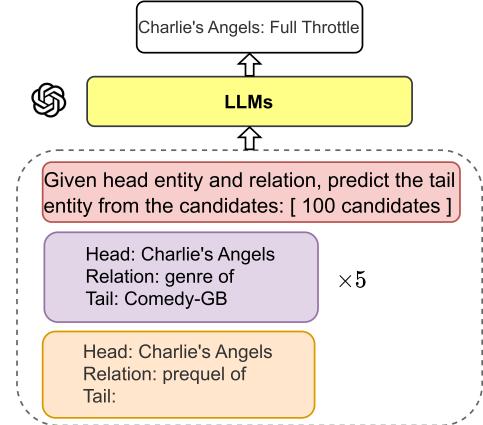


Fig. 16. Framework of prompt-based PaG for KG Completion.

to KGE, conventional KGC methods mainly focused on the structure of the KG, without considering the extensive textual information. However, the recent integration of LLMs enables KGC methods to encode text or generate facts for better KGC performance. These methods fall into two distinct categories based on their utilization styles: 1) *LLM as Encoders (PaE)*, and 2) *LLM as Generators (PaG)*.

**LLM as Encoders (PaE):** This line of work [26], [140], [141], [142], [143] first uses encoder-only LLMs to encode textual information as well as KG facts. Then, they predict the plausibility of the triples or masked entities by feeding the encoded representation into a prediction head, which could be a simple MLP or conventional KG score function (e.g., TransE [33] and TransR [144]).

**LLM as Generators (PaG):** Recent works use LLMs as sequence-to-sequence generators in KGC [96], [145], [146]. These approaches involve encoder-decoder or decoder-only LLMs. The LLMs receive a sequence text input of the query triple ( $h, r, ?$ ), and generate the text of tail entity  $t$  directly. For closed-source LLMs (e.g., ChatGPT and GPT-4), AutoKG adopts prompt engineering to design customized prompts [93]. As shown in Fig. 16, these prompts contain the task description, few-shot examples, and test input, which instruct LLMs to predict the tail entity for KG completion.

More details about the LLM-augmented KG completion can be found in the Appendix B.2, available online.

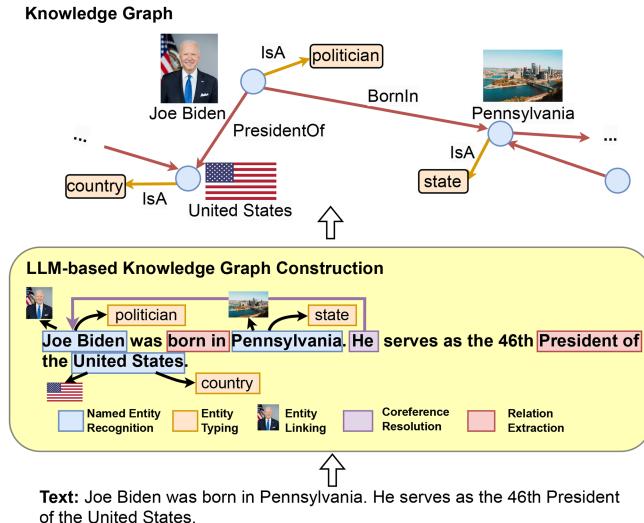


Fig. 17. General framework of LLM-based KG construction.

**Comparison between PaE and PaG:** LLMs as Encoders (PaE) applies an additional prediction head on the top of the representation encoded by LLMs. Therefore, the PaE framework is much easier to finetune since we can only optimize the prediction heads and freeze the LLMs. Moreover, the output of the prediction can be easily specified and integrated with existing KGC functions for different KGC tasks. However, during the inference stage, the PaE requires to compute a score for every candidate in KGs, which could be computationally expensive. Besides, they cannot generalize to unseen entities. Furthermore, the PaE requires the representation output of the LLMs, whereas some state-of-the-art LLMs (e.g. GPT-4<sup>1</sup>) are closed sources and do not grant access to the representation output.

LLMs as Generators (PaG), on the other hand, which does not need the prediction head, can be used without finetuning or access to representations. Therefore, the framework of PaG is suitable for all kinds of LLMs. In addition, PaG directly generates the tail entity, making it efficient in inference without ranking all the candidates and easily generalizing to unseen entities. But, the challenge of PaG is that the generated entities could be diverse and not lie in KGs. What is more, the time of a single inference is longer due to the auto-regressive generation. Last, how to design a powerful prompt that feeds KGs into LLMs is still an open question. Consequently, while PaG has demonstrated promising results for KGC tasks, the trade-off between model complexity and computational efficiency must be carefully considered when selecting an appropriate LLM-based KGC framework.

### C. LLM-Augmented KG Construction

Knowledge graph construction involves creating a structured representation of knowledge within a specific domain. This includes identifying entities and their relationships with each other. The process of knowledge graph construction typically involves multiple stages, including 1) *entity discovery* [147], [148], [149], [150], 2) *coreference resolution* [151], [152], [153], and 3) *relation extraction* [154], [155], [156]. Fig 17 presents

the general framework of applying LLMs for each stage in KG construction. More details about each sub-task are presented in the Appendix B.3.1, B.3.2, available online, and B.3.3, available online, respectively.

Recent approaches have explored 4) *end-to-end knowledge graph construction*, which involves constructing a complete knowledge graph in one step or directly 5) *distilling knowledge graphs from LLMs*.

**End-to-End KG Construction:** Currently, researchers are exploring the use of LLMs for end-to-end KG construction. Kumar et al. [95] propose a unified approach to build KGs from raw text, which contains two LLMs powered components. They first finetune a LLM on named entity recognition tasks to make it capable of recognizing entities in raw text. Then, they propose another “2-model BERT” for solving the relation extraction task, which contains two BERT-based classifiers. The first classifier learns the relation class whereas the second binary classifier learns the direction of the relations between the two entities. The predicted triples and relations are then used to construct the KG. Guo et al. [157] propose an end-to-end knowledge extraction model based on BERT, which can be applied to construct KGs from Classical Chinese text. Grapher [41] presents a novel end-to-end multi-stage system. It first utilizes LLMs to generate KG entities, followed by a simple relation construction head, enabling efficient KG construction from the textual description. PiVE [158] proposes a prompting with an iterative verification framework that utilizes a smaller LLM like T5 to correct the errors in KGs generated by a larger LLM (e.g., ChatGPT). To further explore advanced LLMs, AutoKG design several prompts for different KG construction tasks (e.g., entity typing, entity linking, and relation extraction). Then, it adopts the prompt to perform KG construction using ChatGPT and GPT-4.

**Distilling Knowledge Graphs from LLMs:** LLMs have been shown to implicitly encode massive knowledge [14]. Some research aims to distill knowledge from LLMs to construct KGs. For example, COMET [159] proposes a commonsense transformer model that constructs commonsense KGs by using existing tuples as a seed set of knowledge on which to train. Using this seed set, a LLM learns to adapt its learned representations to knowledge generation, and produces novel tuples that are high quality. Experimental results reveal that implicit knowledge from LLMs is transferred to generate explicit knowledge in commonsense KGs. More details can be found in the Appendix B.3.4, available online.

### D. LLM-Augmented KG-to-Text Generation

The goal of Knowledge-graph-to-text (KG-to-text) generation is to generate high-quality texts that accurately and consistently describe the input knowledge graph information [160]. KG-to-text generation connects knowledge graphs and texts, significantly improving the applicability of KG in more realistic NLG scenarios, including storytelling [161] and knowledge-grounded dialogue [162]. However, it is challenging and costly to collect large amounts of graph-text parallel data, resulting in insufficient training and poor generation quality. Thus, many research efforts resort to either: 1) leverage knowledge from

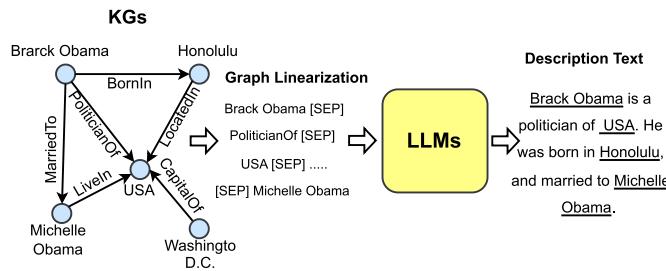


Fig. 18. General framework of KG-to-text generation.

LLMs or 2) construct large-scale weakly-supervised KG-text corpus to solve this issue.

*Leveraging Knowledge from LLMs:* As pioneering research efforts in using LLMs for KG-to-Text generation, Ribeiro et al. [163] and Kale and Rastogi [164] directly fine-tune various LLMs, including BART and T5, with the goal of transferring LLMs knowledge for this task. As shown in Fig. 18, both works simply represent the input graph as a linear traversal and find that such a naive approach successfully outperforms many existing state-of-the-art KG-to-text generation systems.

*Constructing large weakly KG-text aligned Corpus:* Although LLMs have achieved remarkable empirical success, their unsupervised pre-training objectives are not necessarily aligned well with the task of KG-to-text generation, motivating researchers to develop large-scale KG-text aligned corpus. Jin et al. [165] propose a 1.3 M unsupervised KG-to-graph training data from Wikipedia. Similarly, Chen et al. [166] also propose a KG-grounded text corpus collected from the English Wikidump.

More details about the LLM-augmented KG-to-text generation can be found in the Appendix B.4, available online.

### E. LLM-Augmented KG Question Answering

Knowledge graph question answering (KGQA) aims to find answers to natural language questions based on the structured facts stored in knowledge graphs [167], [168]. The inevitable challenge in KGQA is to retrieve related facts and extend the reasoning advantage of KGs to QA. Therefore, recent studies adopt LLMs to bridge the gap between natural language questions and structured knowledge graphs [169], [170], [171], where LLMs can be used as 1) entity/relation extractors, and 2) answer reasoners.

*LLMs as Entity/relation Extractors:* Entity/relation extractors are designed to identify entities and relationships mentioned in natural language questions and retrieve related facts in KGs. Given the proficiency in language comprehension, LLMs can be effectively utilized for this purpose. Lukovnikov et al. [172] are the first to utilize LLMs as classifiers for relation prediction, resulting in a notable improvement in performance compared to shallow neural networks. Nan et al. [171] introduce two LLM-based KGQA frameworks that adopt LLMs to detect mentioned entities and relations. Then, they query the answer in KGs using the extracted entity-relation pairs. QA-GNN [131] uses LLMs to encode the question and candidate answer pairs, which are adopted to estimate the importance of relative KG entities. The entities are retrieved to form a subgraph, where an answer reasoning is conducted by a graph neural network.

TABLE III  
SUMMARY OF METHODS THAT SYNERGIZE KGs AND LLMs

Task	Method	Year
Synergized Knowledge representation	JointGT [42] KEPLER [40] DRAGON [44] HKLM [176]	2021 2021 2022 2023
Synergized Reasoning	LARK [45] Siyuan et al. [46] KSL [177] StructGPT [175] Think-on-graph [178]	2023 2023 2023 2023 2023

*LLMs as Answer Reasoners:* Answer reasoners are designed to reason over the retrieved facts and generate answers. LLMs can be used as answer reasoners to generate answers directly. DEKCOR [169] concatenates the retrieved facts with questions and candidate answers as sentences. Then, it feeds them into LLMs to predict answer scores. GreaseLM [173] fuses the representations from LLMs and graph neural networks to effectively reason over KG facts and language context. UniKGQA [43] unifies the facts retrieval and reasoning into a unified framework. ReLMKG [174] performs joint reasoning on a large language model and the associated knowledge graph. StructGPT [175] adopts a customized interface to allow large language models (e.g., ChatGPT) directly reasoning on KGs to perform multi-step question answering.

More details about LLM-augmented KG question answering can be found in the Appendix B.5, available online.

## VI. SYNERGIZED LLMs + KGs

The synergy of LLMs and KGs has attracted increasing attention these years, which marries the merits of LLMs and KGs to mutually enhance performance in various downstream applications. For example, LLMs can be used to understand natural language, while KGs are treated as a knowledge base, which provides factual knowledge. The unification of LLMs and KGs could result in a powerful model for knowledge representation and reasoning.

In this section, we will discuss the state-of-the-art *Synergized LLMs + KGs* from two perspectives: 1) *Synergized Knowledge Representation*, and 2) *Synergized Reasoning*. Representative works are summarized in Table III.

### A. Synergized Knowledge Representation

Text corpus and knowledge graphs both contain enormous knowledge. However, the knowledge in text corpus is usually implicit and unstructured, while the knowledge in KGs is explicit and structured. Synergized Knowledge Representation aims to design a synergized model that can effectively represent knowledge from both LLMs and KGs. The synergized model can provide a better understanding of the knowledge from both sources, making it valuable for many downstream tasks.

To jointly represent the knowledge, researchers propose the synergized models by introducing additional KG fusion modules, which are jointly trained with LLMs. As shown in Fig. 19, ERNIE [35] proposes a textual-knowledge dual encoder architecture where a *T-encoder* first encodes the input sentences,

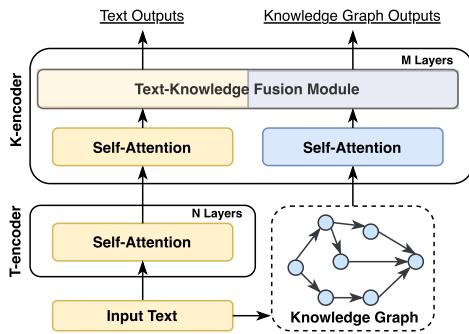


Fig. 19. Synergized knowledge representation by additional KG fusion modules.

then a *K-encoder* processes knowledge graphs which are fused them with the textual representation from the *T-encoder*. BERT-MK [179] employs a similar dual-encoder architecture but it introduces additional information of neighboring entities in the knowledge encoder component during the pre-training of LLMs. However, some of the neighboring entities in KGs may not be relevant to the input text, resulting in extra redundancy and noise. CokeBERT [180] focuses on this issue and proposes a GNN-based module to filter out irrelevant KG entities using the input text. JAKET [181] proposes to fuse the entity information in the middle of the large language model.

KEPLER [40] presents a unified model for knowledge embedding and pre-trained language representation. In KEPLER, they encode textual entity descriptions with a LLM as their embeddings, and then jointly optimize the knowledge embedding and language modeling objectives. JointGT [42] proposes a graph-text joint representation learning model, which proposes three pre-training tasks to align representations of graph and text. DRAGON [44] presents a self-supervised method to pre-train a joint language-knowledge foundation model from text and KG. It takes text segments and relevant KG subgraphs as input and bidirectionally fuses information from both modalities. Then, DRAGON utilizes two self-supervised reasoning tasks, i.e., masked language modeling and KG link prediction to optimize the model parameters. HKLM [176] introduces a unified LLM which incorporates KGs to learn representations of domain-specific knowledge.

### B. Synergized Reasoning

To better utilize the knowledge from text corpus and knowledge graph reasoning, Synergized Reasoning aims to design a synergized model that can effectively conduct reasoning with both LLMs and KGs.

**LLM-KG Fusion Reasoning:** LLM-KG Fusion Reasoning leverages two separated LLM and KG encoders to process the text and relevant KG inputs [182]. These two encoders are equally important and jointly fusing the knowledge from two sources for reasoning. To improve the interaction between text and knowledge, KagNet [38] proposes to first encode the input KG, and then augment the input textual representation. In contrast, MHGRN [168] uses the final LLM outputs of the input text to guide the reasoning process on the KGs. Yet, both of them only design a single-direction interaction between the

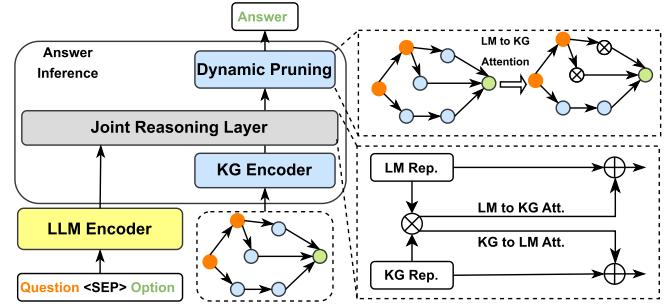


Fig. 20. Framework of LLM-KG Fusion Reasoning.

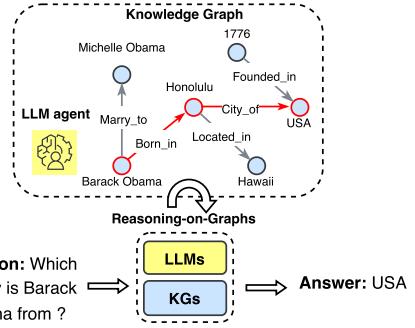


Fig. 21. Using LLMs as agents for reasoning on KGs.

text and KGs. To tackle this issue, QA-GNN [131] proposes to use a GNN-based model to jointly reason over input context and KG information via message passing. Specifically, QA-GNN represents the input textual information as a special node via a pooling operation and connects this node with other entities in KG. However, the textual inputs are only pooled into a single dense vector, limiting the information fusion performance. JointLK [183] then proposes a framework with fine-grained interaction between any tokens in the textual inputs and any KG entities through LM-to-KG and KG-to-LM bi-directional attention mechanism. As shown in Fig. 20, pairwise dot-product scores are calculated over all textual tokens and KG entities, the bi-directional attentive scores are computed separately. In addition, at each jointLK layer, the KGs are also dynamically pruned based on the attention score to allow later layers to focus on more important sub-KG structures. Despite being effective, in JointLK, the fusion process between the input text and KG still uses the final LLM outputs as the input text representations. GreaseLM [173] designs deep and rich interaction between the input text tokens and KG entities at each layer of the LLMs. The architecture and fusion approach is mostly similar to ERNIE [35] discussed in Section VI-A, except that GreaseLM does not use the text-only *T-encoder* to handle input text.

**LLMs as Agents Reasoning:** Instead using two encoders to fuse the knowledge, LLMs can also be treated as agents to interact with the KGs to conduct reasoning [184], as illustrated in Fig. 21. KD-Cot [185] iteratively retrieves facts from KGs and produces faithful reasoning traces, which guide LLMs to generate answers. KSL [177] teaches LLMs to search on KGs to retrieve relevant facts and then generate answers. Struct-GPT [175] designs several API interfaces to allow LLMs to access the structural data and perform reasoning by traversing on KGs. Think-on-graph [178] provides a flexible plug-and-play

framework where LLM agents iteratively execute beam searches on KGs to discover the reasoning paths and generate answers. To enhance the agent abilities, AgentTuning [186] presents several instruction-tuning datasets to guide LLM agents to perform reasoning on KGs.

*Comparison and Discussion:* LLM-KG Fusion Reasoning combines the LLM encoder and KG encoder to represent knowledge in a unified manner. It then employs a synergized reasoning module to jointly reason the results. This framework allows for different encoders and reasoning modules, which are trained end-to-end to effectively utilize the knowledge and reasoning capabilities of LLMs and KGs. However, these additional modules may introduce extra parameters and computational costs while lacking interpretability. LLMs as Agents for KG reasoning provides a flexible framework for reasoning on KGs without additional training cost, which can be generalized to different LLMs and KGs. Meanwhile, the reasoning process is interpretable, which can be used to explain the results. Nevertheless, defining the actions and policies for LLM agents is also challenging. The synergy of LLMs and KGs is still an ongoing research topic, with the potential to have more powerful frameworks in the future.

## VII. FUTURE DIRECTIONS AND MILESTONES

In this section, we discuss the future directions and several milestones in the research area of unifying KGs and LLMs.

### A. KGs for Hallucination Detection in LLMs

The hallucination problem in LLMs, which generates factually incorrect content, significantly hinders the reliability of LLMs. As discussed in Section IV, existing studies try to utilize KGs to obtain more reliable LLMs through pre-training or KG-enhanced inference. Despite the efforts, the issue of hallucination may continue to persist in the realm of LLMs for the foreseeable future. Consequently, in order to gain the public's trust and border applications, it is imperative to detect and assess instances of hallucination within LLMs and other forms of AI-generated content (AIGC). Existing methods strive to detect hallucination by training a neural classifier on a small set of documents [187], which are neither robust nor powerful to handle ever-growing LLMs. Recently, researchers try to use KGs as an external source to validate LLMs [188]. Further studies combine LLMs and KGs to achieve a generalized fact-checking model that can detect hallucinations across domains [189]. Therefore, it opens a new door to utilizing KGs for hallucination detection.

### B. KGs for Editing Knowledge in LLMs

Although LLMs are capable of storing massive real-world knowledge, they cannot quickly update their internal knowledge updated as real-world situations change. There are some research efforts proposed for editing knowledge in LLMs [190] without re-training the whole LLMs. Yet, such solutions still suffer from poor performance or computational overhead [191]. Existing studies [192] also reveal that edit a single fact would cause a ripple effect for other related knowledge. Therefore, it is necessary to develop a more efficient and effective method to

edit knowledge in LLMs. Recently, researchers try to leverage KGs to edit knowledge in LLMs efficiently.

### C. KGs for Black-Box LLMs Knowledge Injection

Although pre-training and knowledge editing could update LLMs to catch up with the latest knowledge, they still need to access the internal structures and parameters of LLMs. However, many state-of-the-art large LLMs (e.g., ChatGPT) only provide APIs for users and developers to access, making themselves black-box to the public. Consequently, it is impossible to follow conventional KG injection approaches described [38], [182] that change LLM structure by adding additional knowledge fusion modules. Converting various types of knowledge into different text prompts seems to be a feasible solution. However, it is unclear whether these prompts can generalize well to new LLMs. Moreover, the prompt-based approach is limited to the length of input tokens of LLMs. Therefore, how to enable effective knowledge injection for black-box LLMs is still an open question for us to explore [193], [194].

### D. Multi-Modal LLMs for KGs

Current knowledge graphs typically rely on textual and graph structure to handle KG-related applications. However, real-world knowledge graphs are often constructed by data from diverse modalities [99], [195], [196]. Therefore, effectively leveraging representations from multiple modalities would be a significant challenge for future research in KGs [197]. One potential solution is to develop methods that can accurately encode and align entities across different modalities. Recently, with the development of multi-modal LLMs [98], [198], leveraging LLMs for modality alignment holds promise in this regard. But, bridging the gap between multi-modal LLMs and KG structure remains a crucial challenge in this field, demanding further investigation and advancements.

### E. LLMs for Understanding KG Structure

Conventional LLMs trained on plain text data are not designed to understand structured data like knowledge graphs. Thus, LLMs might not fully grasp or understand the information conveyed by the KG structure. A straightforward way is to linearize the structured data into a sentence that LLMs can understand. However, the scale of the KGs makes it impossible to linearize the whole KGs as input. Moreover, the linearization process may lose some underlying information in KGs. Therefore, it is necessary to develop LLMs that can directly understand the KG structure and reason over it [175].

### F. Synergized LLMs and KGs for Bidirectional Reasoning

KGs and LLMs are two complementary technologies that can synergize each other. However, the synergy of LLMs and KGs is less explored by existing researchers. A desired synergy of LLMs and KGs would involve leveraging the strengths of both technologies to overcome their individual limitations. LLMs, such as ChatGPT, excel in generating human-like text and understanding natural language, while KGs are structured databases that capture and represent knowledge in a structured manner.

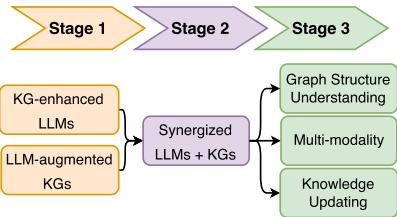


Fig. 22. Milestones of unifying KGs and LLMs.

By combining their capabilities, we can create a powerful system that benefits from the contextual understanding of LLMs and the structured knowledge representation of KGs. To better unify LLMs and KGs, many advanced techniques need to be incorporated, such as multi-modal learning [199], graph neural network [200], and continuous learning [201]. Last, the synergy of LLMs and KGs can be applied to many real-world applications, such as search engines [100], recommender systems [10], [89], and drug discovery.

With a given application problem, we can apply a KG to perform a knowledge-driven search for potential goals and unseen data, and simultaneously start with LLMs to perform a data/text-driven inference to see what new data/goal items can be derived. When the knowledge-based search is combined with data/text-driven inference, they can mutually validate each other, resulting in efficient and effective solutions powered by dual-driving wheels. Therefore, we can anticipate increasing attention to unlock the potential of integrating KGs and LLMs for diverse downstream applications with both generative and reasoning capabilities in the near future.

### VIII. CONCLUSION

Unifying large language models (LLMs) and knowledge graphs (KGs) is an active research direction that has attracted increasing attention from both academia and industry. In this article, we provide a thorough overview of the recent research in this field. We first introduce different manners that integrate KGs to enhance LLMs. Then, we introduce existing methods that apply LLMs for KGs and establish taxonomy based on varieties of KG tasks. Finally, we discuss the challenges and future directions in this field. We envision that there will be multiple stages (milestones) in the roadmap of unifying KGs and LLMs, as shown in Fig. 22. In particular, we will anticipate increasing research on three stages: *Stage 1*: KG-enhanced LLMs, LLM-augmented KGs, *Stage 2*: Synergized LLMs + KGs, and *Stage 3*: Graph Structure Understanding, Multi-modality, Knowledge Updating. We hope that this article will provide a guideline to advance future research.

### REFERENCES

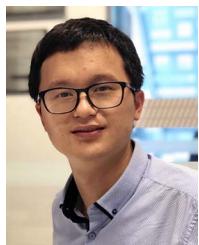
- [1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” 2018, *arXiv: 1810.04805*.
- [2] Y. Liu et al., “RoBERTa: A robustly optimized bert pretraining approach,” 2019, *arXiv: 1907.11692*.
- [3] C. Raffel et al., “Exploring the limits of transfer learning with a unified text-to-text transformer,” *J. Mach. Learn. Res.*, vol. 21, no. 1, pp. 5485–5551, 2020.
- [4] D. Su et al., “Generalizing question answering system with pre-trained language model fine-tuning,” in *Proc. 2nd Workshop Mach. Reading Question Answering*, 2019, pp. 203–211.
- [5] M. Lewis et al., “BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension,” in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 7871–7880.
- [6] J. Li, T. Tang, W. X. Zhao, and J.-R. Wen, “Pretrained language models for text generation: A survey,” 2021, *arXiv:2105.10311*.
- [7] J. Wei et al., “Emergent abilities of large language models,” *Trans. Mach. Learn. Res.*, 2022. [Online]. Available: <https://openreview.net/forum?id=yzkSU5zdwD>
- [8] K. Malinka, M. Perešník, A. Firc, O. Hujník, and F. Januš, “On the educational impact of ChatGPT: Is artificial intelligence ready to obtain a university degree?,” 2023, *arXiv:2303.11146*.
- [9] Z. Li, C. Wang, Z. Liu, H. Wang, S. Wang, and C. Gao, “CCTest: Testing and repairing code completion systems,” in *Proc. IEEE/ACM Int. Conf. Softw. Eng.*, 2023, pp. 1238–1250.
- [10] J. Liu, C. Liu, R. Lv, K. Zhou, and Y. Zhang, “Is ChatGPT a good recommender? A preliminary study,” 2023, *arXiv:2304.10149*.
- [11] W. X. Zhao et al., “A survey of large language models,” 2023, *arXiv:2303.18223*.
- [12] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang, “Pre-trained models for natural language processing: A survey,” *Sci. China Technological Sci.*, vol. 63, no. 10, pp. 1872–1897, 2020.
- [13] J. Yang et al., “Harnessing the power of LLMs in practice: A survey on ChatGPT and beyond,” 2023, *arXiv:2304.13712*.
- [14] F. Petroni et al., “Language models as knowledge bases?,” in *Proc. Conf. Empir. Methods Natural Lang. Joint Conf. Natural Lang. Process.*, 2019, pp. 2463–2473.
- [15] Z. Ji et al., “Survey of hallucination in natural language generation,” *ACM Comput. Surv.*, vol. 55, no. 12, pp. 1–38, 2023.
- [16] H. Zhang, H. Song, S. Li, M. Zhou, and D. Song, “A survey of controllable text generation using transformer-based pre-trained language models,” 2022, *arXiv:2201.05337*.
- [17] M. Danilevsky, K. Qian, R. Aharonov, Y. Katsis, B. Kawas, and P. Sen, “A survey of the state of explainable AI for natural language processing,” 2020, *arXiv: 2010.00711*.
- [18] J. Wang et al., “On the robustness of ChatGPT: An adversarial and out-of-distribution perspective,” 2023, *arXiv:2302.12095*.
- [19] S. Ji, S. Pan, E. Cambria, P. Marttinen, and S. Y. Philip, “A survey on knowledge graphs: Representation, acquisition, and applications,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 2, pp. 494–514, Feb. 2022.
- [20] D. Vrandečić and M. Krötzsch, “Wikidata: A free collaborative knowledgebase,” *Commun. ACM*, vol. 57, no. 10, pp. 78–85, 2014.
- [21] S. Hu, L. Zou, and X. Zhang, “A state-transition framework to answer complex questions over knowledge base,” in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 2098–2108.
- [22] J. Zhang, B. Chen, L. Zhang, X. Ke, and H. Ding, “Neural, symbolic and neural-symbolic reasoning on knowledge graphs,” *AI Open*, vol. 2, pp. 14–35, 2021.
- [23] B. Abu-Salih, “Domain-specific knowledge graphs: A survey,” *J. Netw. Comput. Appl.*, vol. 185, 2021, Art. no. 103076.
- [24] T. Mitchell et al., “Never-ending learning,” *Commun. ACM*, vol. 61, no. 5, pp. 103–115, 2018.
- [25] L. Zhong, J. Wu, Q. Li, H. Peng, and X. Wu, “A comprehensive survey on automatic knowledge graph construction,” 2023, *arXiv:2302.05019*.
- [26] L. Yao, C. Mao, and Y. Luo, “KG-BERT: BERT for knowledge graph completion,” 2019, *arXiv: 1909.03193*.
- [27] L. Luo, Y.-F. Li, G. Haffari, and S. Pan, “Normalizing flow-based neural process for few-shot knowledge graph completion,” in *Proc. 46th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2023, pp. 900–910.
- [28] Y. Bang et al., “A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity,” 2023, *arXiv:2302.04023*.
- [29] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, and D. Zhou, “Self-consistency improves chain of thought reasoning in language models,” 2022, *arXiv:2203.11171*.
- [30] O. Golovneva et al., “ROSCOE: A suite of metrics for scoring step-by-step reasoning,” in *Proc. 11th Int. Conf. Learn. Representations*, 2023.
- [31] F. M. Suchanek, G. Kasneci, and G. Weikum, “YAGO: A core of semantic knowledge,” in *Proc. World Wide Web Conf.*, 2007, pp. 697–706.
- [32] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. Hruschka, and T. Mitchell, “Toward an architecture for never-ending language learning,” in *Proc. AAAI Conf. Artif. Intell.*, 2010, pp. 1306–1313.

- [33] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, “Translating embeddings for modeling multi-relational data,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2013.
- [34] G. Wan, S. Pan, C. Gong, C. Zhou, and G. Haffari, “Reasoning like human: Hierarchical reinforcement learning for knowledge graph reasoning,” in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 1926–1932.
- [35] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, “ERNIE: Enhanced language representation with informative entities,” in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 1441–1451.
- [36] W. Liu et al., “K-BERT: Enabling language representation with knowledge graph,” in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 2901–2908.
- [37] Y. Liu, Y. Wan, L. He, H. Peng, and P. S. Yu, “KG-BART: Knowledge graph-augmented BART for generative commonsense reasoning,” in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 6418–6425.
- [38] B. Y. Lin, X. Chen, J. Chen, and X. Ren, “KagNet: Knowledge-aware graph networks for commonsense reasoning,” in *Proc. Conf. Empir. Methods Natural Lang. Joint Conf. Natural Lang. Process.*, 2019, pp. 2829–2839.
- [39] D. Dai, L. Dong, Y. Hao, Z. Sui, B. Chang, and F. Wei, “Knowledge neurons in pretrained transformers,” 2021, *arXiv:2104.08696*.
- [40] X. Wang et al., “KEPLER: A unified model for knowledge embedding and pre-trained language representation,” *Trans. Assoc. Comput. Linguistics*, vol. 9, pp. 176–194, 2021.
- [41] I. Melnyk, P. Dognin, and P. Das, “Grapher: Multi-stage knowledge graph construction using pretrained language models,” in *Proc. NeurIPS 2021 Workshop Deep Generative Models Downstream Appl.*, 2021.
- [42] P. Ke et al., “JointGT: Graph-text joint representation learning for text generation from knowledge graphs,” in *Proc. Annu. Meeting Assoc. Comput. Linguistics Finding*, 2021, pp. 2526–2538.
- [43] J. Jiang, K. Zhou, W. X. Zhao, and J.-R. Wen, “UniKGQA: Unified retrieval and reasoning for solving multi-hop question answering over knowledge graph,” in *Proc. 11th Int. Conf. Learn. Representations*, 2023.
- [44] M. Yasunaga et al., “Deep bidirectional language-knowledge graph pretraining,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2022, pp. 37 309–37 323.
- [45] N. Choudhary and C. K. Reddy, “Complex logical reasoning over knowledge graphs using large language models,” 2023, *arXiv:2305.01157*.
- [46] S. Wang, Z. Wei, J. Xu, and Z. Fan, “Unifying structure reasoning and language model pre-training for complex reasoning,” 2023, *arXiv:2301.08913*.
- [47] C. Zhen, Y. Shang, X. Liu, Y. Li, Y. Chen, and D. Zhang, “A survey on knowledge-enhanced pre-trained language models,” 2022, *arXiv:2212.13428*.
- [48] X. Wei, S. Wang, D. Zhang, P. Bhatia, and A. Arnold, “Knowledge enhanced pretrained language models: A comprehensive survey,” 2021, *arXiv:2110.08455*.
- [49] D. Yin et al., “A survey of knowledge-intensive NLP with pre-trained language models,” 2022, *arXiv:2202.08772*.
- [50] A. Vaswani et al., “Attention is all you need,” in *Proc. Adv. Neural Inf. Process. Syst.*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett, Eds., vol. 30, 2017, pp. 6000–6010.
- [51] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, “ALBERT: A lite bert for self-supervised learning of language representations,” in *Proc. Int. Conf. Learn. Representations*, 2020.
- [52] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, “ELECTRA: Pre-training text encoders as discriminators rather than generators,” 2020, *arXiv: 2003.10555*.
- [53] K. Hakala and S. Pyysalo, “Biomedical named entity recognition with multilingual BERT,” in *Proc. 5th Workshop BioNLP Open Shared Tasks*, 2019, pp. 56–61.
- [54] Y. Tay et al., “UL2: Unifying language learning paradigms,” in *Proc. 11th Int. Conf. Learn. Representations*, 2023.
- [55] V. Sanh et al., “Multitask prompted training enables zero-shot task generalization,” in *Proc. Int. Conf. Learn. Representations*, 2022.
- [56] B. Zoph et al., “ST-MoE: Designing stable and transferable sparse expert models,” 2022. [Online]. Available: <https://arxiv.org/abs/2202.08906>
- [57] A. Zeng et al., “GLM-130B: An open bilingual pre-trained model,” in *Proc. 11th Int. Conf. Learn. Representations*, 2023. [Online]. Available: <https://openreview.net/forum?id=-Aw0rrrPUF>
- [58] L. Xue et al., “mT5: A massively multilingual pre-trained text-to-text transformer,” in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2021, pp. 483–498.
- [59] T. Brown et al., “Language models are few-shot learners,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 1877–1901.
- [60] L. Ouyang et al., “Training language models to follow instructions with human feedback,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2022, pp. 27 730–27 744.
- [61] H. Touvron et al., “LLaMA: Open and efficient foundation language models,” 2023, *arXiv:2302.13971*.
- [62] E. Saravia, “Prompt engineering guide,” 2022, Accessed: Dec. 2022. <https://github.com/dair-ai/Prompt-Engineering-Guide>
- [63] J. Wei et al., “Chain-of-thought prompting elicits reasoning in large language models,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 24824–24837.
- [64] S. Li et al., “Graph reasoning for question answering with triplet retrieval,” in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2023, pp. 3366–3375.
- [65] Y. Wen, Z. Wang, and J. Sun, “MindMap: Knowledge graph prompting sparks graph of thoughts in large language models,” 2023, *arXiv:2308.09729*.
- [66] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, “Freebase: A collaboratively created graph database for structuring human knowledge,” in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2008, pp. 1247–1250.
- [67] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, “DBpedia: A nucleus for a web of open data,” in *Proc. 6th Int. Semantic Web Conf.*, 2007, pp. 722–735.
- [68] B. Xu et al., “CN-DBpedia: A never-ending Chinese knowledge extraction system,” in *Proc. 30th Int. Conf. Ind. Eng. Other Appl. Appl. Intell. Syst.*, 2017, pp. 428–438.
- [69] P. Hai-Nyzhnyk, “Vikidia as a universal multilingual online encyclopedia for children,” *Encyclopedia Herald Ukraine*, vol. 14, pp. 81–87, 2022.
- [70] F. Ilievski, P. Szekely, and B. Zhang, “CSKG: The commonsense knowledge graph,” in *Proc. Extended Semantic Web Conf.*, 2021.
- [71] R. Speer, J. Chin, and C. Havasi, “ConceptNet 5.5: An open multilingual graph of general knowledge,” in *Proc. AAAI Conf. Artif. Intell.*, vol. 31, no. 1, Feb. 2017, doi: [10.1109/aaai.v31i1.11164](https://doi.org/10.1109/aaai.v31i1.11164). [Online]. Available: <https://ojs.aaai.org/index.php/AAAI/article/view/11164>
- [72] H. Ji, P. Ke, S. Huang, F. Wei, X. Zhu, and M. Huang, “Language generation with multi-hop reasoning on commonsense knowledge graph,” in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2020, pp. 725–736.
- [73] J. D. Hwang et al., “(Comet-) atomic 2020: On symbolic and neural commonsense knowledge graphs,” in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 7, 2021, pp. 6384–6392.
- [74] H. Zhang, X. Liu, H. Pan, Y. Song, and C. W.-K. Leung, “ASER: A large-scale eventuality knowledge graph,” in *Proc. Web Conf.*, 2020, pp. 201–211.
- [75] H. Zhang, D. Khashabi, Y. Song, and D. Roth, “TransOMCS: From linguistic graphs to commonsense knowledge,” in *Proc. Int. Joint Conf. Artif. Intell.*, 2021, pp. 4004–4010.
- [76] Z. Li, X. Ding, T. Liu, J. E. Hu, and B. Van Durme, “Guided generation of cause and effect,” in *Proc. Int. Joint Conf. Artif. Intell.*, 2020.
- [77] O. Bodenreider, “The unified medical language system (UMLS): Integrating biomedical terminology,” *Nucleic acids Res.*, vol. 32, no. suppl\_1, pp. D267–D270, 2004.
- [78] Y. Liu, Q. Zeng, J. Ordieres Meré, and H. Yang, “Anticipating stock market of the renowned companies: A knowledge graph approach,” *Complexity*, vol. 2019, 2019, Art. no. 9202457.
- [79] Y. Zhu et al., “Intelligent learning for knowledge graph towards geological data,” *Sci. Program.*, vol. 2017, 2017, Art. no. 5072427.
- [80] W. Choi and H. Lee, “Inference of biomedical relations among chemicals, genes, diseases, and symptoms using knowledge representation learning,” *IEEE Access*, vol. 7, pp. 179 373–179 384, 2019.
- [81] F. Farazi et al., “Knowledge graph approach to combustion chemistry and interoperability,” *ACS Omega*, vol. 5, no. 29, pp. 18 342–18 348, 2020.
- [82] X. Wu, T. Jiang, Y. Zhu, and C. Bu, “Knowledge graph for China’s genealogy,” *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 634–646, Jan. 2023.
- [83] X. Zhu et al., “Multi-modal knowledge graph construction and application: A survey,” *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 2, pp. 715–735, 2024, doi: [10.1109/TKDE.2022.3224228](https://doi.org/10.1109/TKDE.2022.3224228).
- [84] S. Ferrada, B. Bustos, and A. Hogan, “IMGpedia: A linked dataset with content-based analysis of WIKIMEDIA images,” in *Proc. Int. Semantic Web Conf.*, 2017, pp. 84–93.

- [85] Y. Liu, H. Li, A. Garcia-Duran, M. Niepert, D. Onoro-Rubio, and D. S. Rosenblum, "MMKG: Multi-modal knowledge graphs," in *Proc. 16th Int. Conf. Semantic Web*, 2019, pp. 459–474.
- [86] M. Wang, H. Wang, G. Qi, and Q. Zheng, "Richpedia: A large-scale, comprehensive multi-modal knowledge graph," *Big Data Res.*, vol. 22, 2020, Art. no. 100159.
- [87] B. Shi, L. Ji, P. Lu, Z. Niu, and N. Duan, "Knowledge aware semantic concept expansion for image-text matching," in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, Art. no. 2.
- [88] S. Shah, A. Mishra, N. Yadati, and P. P. Talukdar, "KVQA: Knowledge-aware visual question answering," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 01, 2019, pp. 8876–8884.
- [89] R. Sun et al., "Multi-modal knowledge graphs for recommender systems," in *Proc. Conf. Inf. Knowl. Manage.*, 2020, pp. 1405–1414.
- [90] S. Deng et al., "Construction and applications of billion-scale pre-trained multimodal business knowledge graph," in *Proc. IEEE 39th Int. Conf. Data Eng.*, 2023, pp. 2988–3002.
- [91] C. Rosset, C. Xiong, M. Phan, X. Song, P. Bennett, and S. Tiwary, "Knowledge-aware language model pretraining," 2020, *arXiv: 2007.00655*.
- [92] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 9459–9474.
- [93] Y. Zhu et al., "LLMs for knowledge graph construction and reasoning: Recent capabilities and future opportunities," 2023, *arXiv: 2305.13168*.
- [94] Z. Zhang, X. Liu, Y. Zhang, Q. Su, X. Sun, and B. He, "Pretrain-KGE: Learning knowledge representation from pretrained language models," in *Proc. Conf. Empirical Methods Natural Lang. Process. Finding*, 2020, pp. 259–266.
- [95] A. Kumar, A. Pandey, R. Gadia, and M. Mishra, "Building knowledge graph using pre-trained language model for learning entity-aware relationships," in *Proc. IEEE Int. Conf. Comput. Power Commun. Technol.*, 2020, pp. 310–315.
- [96] X. Xie et al., "From discrimination to generation: Knowledge graph completion with generative transformer," in *Proc. World Wide Web Conf.*, 2022, pp. 162–165.
- [97] Z. Chen, C. Xu, F. Su, Z. Huang, and Y. Dou, "Incorporating structured sentences with time-enhanced bert for fully-inductive temporal relation prediction," in *Proc. 46th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2023.
- [98] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny, "MiniGPT-4: Enhancing vision-language understanding with advanced large language models," 2023, *arXiv: 2304.10592*.
- [99] M. Warren, D. A. Shamma, and P. J. Hayes, "Knowledge engineering with image data in real-world settings," in *Proc. AAAI Spring Symp. Combining Mach. Learn. Knowl. Eng.*, Mar. 2021.
- [100] R. Thoppilan et al., "LaMDA: Language models for dialog applications," 2022, *arXiv: 2201.08239*.
- [101] Y. Sun et al., "ERNIE 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation," 2021, *arXiv: 2107.02137*.
- [102] T. Shen, Y. Mao, P. He, G. Long, A. Trischler, and W. Chen, "Exploiting structured knowledge in text via graph-guided representation learning," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2020, pp. 8980–8994.
- [103] D. Zhang, Z. Yuan, Y. Liu, F. Zhuang, H. Chen, and H. Xiong, "E-BERT: A phrase and product knowledge enhanced language model for E-commerce," 2020, *arXiv: 2009.02835*.
- [104] S. Li et al., "Pre-training language models with deterministic factual knowledge," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2022, pp. 11 118–11 131.
- [105] M. Kang, J. Baek, and S. J. Hwang, "KAL: Knowledge-augmented language model adaptation," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2022, pp. 5144–5167.
- [106] W. Xiong, J. Du, W. Y. Wang, and V. Stoyanov, "Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model," in *Proc. Int. Conf. Learn. Representations*, 2020. [Online]. Available: <https://openreview.net/forum?id=BJIzm64tDH>
- [107] T. Sun et al., "CoLAKE: Contextualized language and knowledge embedding," in *Proc. 28th Int. Conf. Comput. Linguistics*, 2020, pp. 3660–3670.
- [108] T. Zhang et al., "DKPLM: Decomposable knowledge-enhanced pre-trained language model for natural language understanding," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 11 703–11 711.
- [109] J. Wang et al., "Knowledge prompting in pre-trained language model for natural language understanding," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2022, pp. 3164–3177.
- [110] H. Ye et al., "Ontology-enhanced prompt-tuning for few-shot learning," in *Proc. ACM Web Conf.*, 2022, pp. 778–787.
- [111] H. Luo et al., "ChatKBQA: A generate-then-retrieve framework for knowledge base question answering with fine-tuned large language models," 2023, *arXiv: 2310.08975*.
- [112] L. Luo, Y.-F. Li, G. Haffari, and S. Pan, "Reasoning on graphs: Faithful and interpretable large language model reasoning," 2023, *arXiv: 2310.01061*.
- [113] R. Logan, N. F. Liu, M. E. Peters, M. Gardner, and S. Singh, "Barack's wife Hillary: Using knowledge graphs for fact-aware language modeling," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 5962–5971.
- [114] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang, "REALM: Retrieval-augmented language model pre-training," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 3929–3938.
- [115] Y. Wu, Y. Zhao, B. Hu, P. Minervini, P. Stenetorp, and S. Riedel, "An efficient memory-augmented transformer for knowledge-intensive NLP tasks," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2022, pp. 5184–5196.
- [116] L. Luo, J. Ju, B. Xiong, Y.-F. Li, G. Haffari, and S. Pan, "ChatRule: Mining logical rules with large language models for knowledge graph reasoning," 2023, *arXiv: 2309.01538*.
- [117] J. Wang, Q. Sun, N. Chen, X. Li, and M. Gao, "Boosting language models reasoning with chain-of-knowledge prompting," 2023, *arXiv: 2306.06427*.
- [118] Z. Jiang, F. F. Xu, J. Araki, and G. Neubig, "How can we know what language models know?," *Trans. Assoc. Comput. Linguistics*, vol. 8, pp. 423–438, 2020.
- [119] T. Shin, Y. Razeghi, R. L. Logan IV, E. Wallace, and S. Singh, "Auto-Prompt: Eliciting knowledge from language models with automatically generated prompts," 2020, *arXiv: 2010.15980*.
- [120] Z. Meng, F. Liu, E. Shareghi, Y. Su, C. Collins, and N. Collier, "Rewire-then-probe: A contrastive recipe for probing biomedical knowledge of pre-trained language models," 2021, *arXiv: 2110.08173*.
- [121] L. Luo, T.-T. Vu, D. Phung, and G. Haffari, "Systematic assessment of factual knowledge in large language models," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2023.
- [122] V. Swamy, A. Romanou, and M. Jaggi, "Interpreting language models through knowledge graph extraction," 2021, *arXiv: 2111.08546*.
- [123] S. Li et al., "How pre-trained language models capture factual knowledge? A causal-inspired analysis," 2022, *arXiv: 2203.16747*.
- [124] H. Tian et al., "SKEP: Sentiment knowledge enhanced pre-training for sentiment analysis," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 4067–4076.
- [125] W. Yu et al., "Dict-BERT: Enhancing language model pre-training with dictionary," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 1907–1918.
- [126] T. McCoy, E. Pavlick, and T. Linzen, "Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 3428–3448.
- [127] D. Wilmot and F. Keller, "Memory and knowledge augmented language models for inferring salience in long-form stories," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2021, pp. 851–865.
- [128] L. Adolphs, S. Dhuliawala, and T. Hofmann, "How to query language models?," 2021, *arXiv: 2108.01928*.
- [129] M. Sung, J. Lee, S. Yi, M. Jeon, S. Kim, and J. Kang, "Can language models be biomedical knowledge bases?," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2021, pp. 4723–4734.
- [130] A. Mallen, A. Asai, V. Zhong, R. Das, H. Hajishirzi, and D. Khashabi, "When not to trust language models: Investigating effectiveness and limitations of parametric and non-parametric memories," 2022, *arXiv: 2212.10511*.
- [131] M. Yasunaga, H. Ren, A. Bosselut, P. Liang, and J. Leskovec, "QA-GNN: Reasoning with language models and knowledge graphs for question answering," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2021, pp. 535–546.
- [132] X. Huang, J. Zhang, D. Li, and P. Li, "Knowledge graph embedding based question answering," in *Proc. ACM Int. Conf. Web Search Data Mining*, 2019, pp. 105–113.
- [133] H. Wang, F. Zhang, X. Xie, and M. Guo, "DKN: Deep knowledge-aware network for news recommendation," in *Proc. World Wide Web Conf.*, 2018, pp. 1835–1844.

- [134] B. Yang, S. W.-T. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," in *Proc. Int. Conf. Learn. Representations*, 2015.
- [135] W. Xiong, M. Yu, S. Chang, X. Guo, and W. Y. Wang, "One-shot relational learning for knowledge graphs," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 1980–1990.
- [136] P. Wang, J. Han, C. Li, and R. Pan, "Logic attention based neighborhood aggregation for inductive knowledge graph embedding," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 7152–7159.
- [137] N. Zhang et al., "Reasoning through memorization: Nearest neighbor knowledge graph embeddings," 2022, *arXiv:2201.05575*.
- [138] X. Wang, Q. He, J. Liang, and Y. Xiao, "Language models as knowledge embeddings," 2022, *arXiv:2206.12617*.
- [139] X. Xie et al., "LambdaKG: A library for pre-trained language model-based knowledge graph embeddings," 2022.
- [140] J. Shen, C. Wang, L. Gong, and D. Song, "Joint language semantic and structure embedding for knowledge graph completion," in *Proc. Int. Conf. Comput. Linguistics*, 2022, pp. 1965–1978.
- [141] B. Choi, D. Jang, and Y. Ko, "MEM-KGC: Masked entity model for knowledge graph completion with pre-trained language model," *IEEE Access*, vol. 9, pp. 132 025–132 032, 2021.
- [142] B. Wang, T. Shen, G. Long, T. Zhou, Y. Wang, and Y. Chang, "Structure-augmented text representation learning for efficient knowledge graph completion," in *Proc. World Wide Web Conf.*, 2021, pp. 1737–1748.
- [143] L. Wang, W. Zhao, Z. Wei, and J. Liu, "SimKGC: Simple contrastive knowledge graph completion with pre-trained language models," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 4281–4294.
- [144] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proc. AAAI Conf. Artif. Intell.*, 2015.
- [145] A. Saxena, A. Kochsiek, and R. Gemulla, "Sequence-to-sequence knowledge graph completion and question answering," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 2814–2828.
- [146] C. Chen, Y. Wang, B. Li, and K. Lam, "Knowledge is flat: A Seq2Seq generative framework for various knowledge graph completion," in *Proc. Int. Conf. Comput. Linguistics*, 2022, pp. 4005–4017.
- [147] M. E. Peters et al., "Deep contextualized word representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2018, pp. 2227–2237.
- [148] Y. Onoe and G. Durrett, "Learning to denoise distantly-labeled data for entity typing," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2019, pp. 2407–2417.
- [149] N. D. Cao, G. Izacard, S. Riedel, and F. Petroni, "Autoregressive entity retrieval," in *Proc. Int. Conf. Learn. Representations*, Austria, 2021, 2021.
- [150] T. Ayoola, S. Tyagi, J. Fisher, C. Christodoulopoulos, and A. Pierleoni, "Refined: An efficient zero-shot-capable approach to end-to-end entity linking," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2022, pp. 209–220.
- [151] M. Joshi, O. Levy, L. Zettlemoyer, and D. S. Weld, "BERT for coreference resolution: Baselines and analysis," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2019, pp. 5802–5807.
- [152] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, "SpanBERT: Improving pre-training by representing and predicting spans," *Trans. Assoc. Comput. Linguistics*, vol. 8, pp. 64–77, 2020.
- [153] A. Cattan, A. Eirw, G. Stanovsky, M. Joshi, and I. Dagan, "Cross-document coreference resolution over predicted mentions," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2021, pp. 5100–5107.
- [154] P. Shi and J. Lin, "Simple BERT models for relation extraction and semantic role labeling," 2019, *arXiv: 1904.05255*.
- [155] C. Alt, M. Hübner, and L. Hennig, "Improving relation extraction by pre-trained language representations," in *Proc. 1st Conf. Automated Knowl. Base Construction*, Amherst, MA, USA, 2019.
- [156] Y. Ma, A. Wang, and N. Okazaki, "DREAM: Guiding attention with evidence for improving document-level relation extraction," in *Proc. 13th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2023, pp. 1963–1975.
- [157] Q. Guo et al., "Constructing chinese historical literature knowledge graph based on BERT," in *Proc. 18th Int. Conf. Web Inf. Syst. Appl.*, Kaifeng, China, 2021, 2021, pp. 323–334.
- [158] J. Han, N. Collier, W. Buntine, and E. Shareghi, "PiVe: Prompting with iterative verification improving graph-based generative capability of LLMs," 2023, *arXiv:2305.12392*.
- [159] A. Bosselut, H. Rashkin, M. Sap, C. Malaviya, A. Celikyilmaz, and Y. Choi, "COMET: Commonsense transformers for knowledge graph construction," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*,
- [160] A. Korhonen, D. Traum, and Luis Márquez, Eds., Florence, Italy: Association for Computational Linguistics, pp. 4762–4779, Jul. 2019, doi: 10.18653/v1/P19-1470. [Online]. Available: <https://aclanthology.org/P19-1470>
- [161] C. Gardent, A. Shimorina, S. Narayan, and L. Perez-Beltrachini, "The WebNLG challenge: Generating text from RDF data," in *Proc. 10th Int. Conf. Natural Lang. Gener.*, 2017, pp. 124–133.
- [162] H. Zhou, T. Young, M. Huang, H. Zhao, J. Xu, and X. Zhu, "Commonsense knowledge aware conversation generation with graph attention," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, pp. 4623–4629.
- [163] L. F. R. Ribeiro, M. Schmitt, H. Schütze, and I. Gurevych, "Investigating pretrained language models for graph-to-text generation," in *Proc. 3rd Workshop Natural Lang. Process. Conversational AI*, 2021, pp. 211–227.
- [164] M. Kale and A. Rastogi, "Text-to-text pre-training for data-to-text tasks," in *Proc. 13th Int. Conf. Natural Lang. Gener.*, 2020, pp. 97–102.
- [165] Z. Jin, Q. Guo, X. Qiu, and Z. Zhang, "GenWiki: A dataset of 1.3 million content-sharing text and graphs for unsupervised graph-to-text generation," in *Proc. 28th Int. Conf. Comput. Linguistics*, 2020, pp. 2398–2409.
- [166] W. Chen, Y. Su, X. Yan, and W. Y. Wang, "KGPT: Knowledge-grounded pre-training for data-to-text generation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2020, pp. 8635–8648.
- [167] A. Saxena, A. Tripathi, and P. Talukdar, "Improving multi-hop question answering over knowledge graphs using knowledge base embeddings," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 4498–4507.
- [168] Y. Feng, X. Chen, B. Y. Lin, P. Wang, J. Yan, and X. Ren, "Scalable multi-hop relational reasoning for knowledge-aware question answering," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2020, pp. 1295–1309.
- [169] Y. Xu, C. Zhu, R. Xu, Y. Liu, M. Zeng, and X. Huang, "Fusing context into knowledge graph for commonsense question answering," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2021, pp. 1201–1207.
- [170] Y. Yan et al., "Large-scale relation learning for question answering over knowledge bases with pre-trained language models," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2021, pp. 3653–3660.
- [171] N. Hu et al., "An empirical study of pre-trained language models in simple knowledge graph question answering," 2023, *arXiv:2303.10368*.
- [172] D. Lukovnikov, A. Fischer, and J. Lehmann, "Pretrained transformers for simple question answering over knowledge graphs," in *The Semantic Web-ISWC in Proc. 18th Int. Semantic Web Conf.*, Auckland, New Zealand, 2019, pp. 470–486.
- [173] X. Zhang et al., "GreaseLM: Graph reasoning enhanced language models," in *Proc. Int. Conf. Learn. Representations*, 2022.
- [174] X. Cao and Y. Liu, "ReLMKG: Reasoning with pre-trained language models and knowledge graphs for complex question answering," *Appl. Intell.*, vol. 53, pp. 12032–12046, 2023.
- [175] J. Jiang, K. Zhou, Z. Dong, K. Ye, W. X. Zhao, and J.-R. Wen, "StructGPT: A general framework for large language model to reason over structured data," 2023, *arXiv:2305.09645*.
- [176] H. Zhu, H. Peng, Z. Lyu, L. Hou, J. Li, and J. Xiao, "Pre-training language model incorporating domain-specific heterogeneous knowledge into a unified representation," *Expert Syst. Appl.*, vol. 215, 2023, Art. no. 119369.
- [177] C. Feng, X. Zhang, and Z. Fei, "Knowledge solver: Teaching LLMs to search for domain knowledge from knowledge graphs," 2023, *arXiv:2309.03118*.
- [178] J. Sun et al., "Think-on-graph: Deep and responsible reasoning of large language model with knowledge graph," 2023, *arXiv:2307.07697*.
- [179] B. He et al., "BERT-MK: Integrating graph contextualized knowledge into pre-trained language models," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2020, pp. 2281–2290.
- [180] Y. Su et al., "CokeBERT: Contextual knowledge selection and embedding towards enhanced pre-trained language models," *AI Open*, vol. 2, pp. 127–134, 2021.
- [181] D. Yu, C. Zhu, Y. Yang, and M. Zeng, "JAKET: Joint pre-training of knowledge graph and language understanding," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 11 630–11 638.
- [182] X. Wang et al., "Improving natural language inference using external knowledge in the science questions domain," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 7208–7215.

- [183] Y. Sun, Q. Shi, L. Qi, and Y. Zhang, "JointLK: Joint reasoning with language models and knowledge graphs for commonsense question answering," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2022, pp. 5049–5060.
- [184] X. Liu et al., "AgentBench: Evaluating LLMs as agents," 2023, *arXiv:2308.03688*.
- [185] Y. Wang, N. Lipka, R. A. Rossi, A. Siu, R. Zhang, and T. Derr, "Knowledge graph prompting for multi-document question answering," 2023, *arXiv:2308.11730*.
- [186] A. Zeng et al., "AgentTuning: Enabling generalized agent abilities for LLMs," 2023, *arXiv:2310.12823*.
- [187] W. Kryściński, B. McCann, C. Xiong, and R. Socher, "Evaluating the factual consistency of abstractive text summarization," 2019, *arXiv:1910.12840*.
- [188] Z. Ji et al., "RHO ( $\setminus \rho$ ): Reducing hallucination in open-domain dialogues with knowledge grounding," 2022, *arXiv:2212.01588*.
- [189] S. Feng, V. Balachandran, Y. Bai, and Y. Tsvetkov, "FactKB: Generalizable factuality evaluation using language models enhanced with factual knowledge," 2023, *arXiv:2305.08281*.
- [190] Y. Yao et al., "Editing large language models: Problems, methods, and opportunities," 2023, *arXiv:2305.13172*.
- [191] Z. Li, N. Zhang, Y. Yao, M. Wang, X. Chen, and H. Chen, "Unveiling the pitfalls of knowledge editing for large language models," 2023, *arXiv:2310.02129*.
- [192] R. Cohen, E. Biran, O. Yoran, A. Globerson, and M. Geva, "Evaluating the ripple effects of knowledge editing in language models," 2023, *arXiv:2307.12976*.
- [193] S. Diao et al., "Black-box prompt learning for pre-trained language models," 2022, *arXiv:2201.08531*.
- [194] T. Sun, Y. Shao, H. Qian, X. Huang, and X. Qiu, "Black-box tuning for language-model-as-a-service," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 20 841–20 855.
- [195] X. Chen, A. Shrivastava, and A. Gupta, "NEIL: Extracting visual knowledge from web data," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sydney, Australia, 2013, pp. 1409–1416.
- [196] M. Warren and P. J. Hayes, "Bounding ambiguity: Experiences with an image annotation system," in *Proc. 1st Workshop Subjectivity Ambiguity Disagreement Crowdsourcing*, 2018, pp. 41–54.
- [197] Z. Chen et al., "LaKo: Knowledge-driven visual estimation answering via late knowledge-to-text injection," in *Proc. 11th Int. Joint Conf. Knowl. Graphs*, 2022, pp. 20–29.
- [198] R. Girdhar et al., "ImageBind: One embedding space to bind them all," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2023, pp. 15 180–15 190.
- [199] J. Zhang, Z. Yin, P. Chen, and S. Nicelle, "Emotion recognition using multi-modal data and machine learning techniques: A tutorial and review," *Inf. Fusion*, vol. 59, pp. 103–126, 2020.
- [200] H. Zhang, B. Wu, X. Yuan, S. Pan, H. Tong, and J. Pei, "Trustworthy graph neural networks: Aspects, methods and trends," 2022, *arXiv:2205.07424*.
- [201] T. Wu, M. Caccia, Z. Li, Y.-F. Li, G. Qi, and G. Haffari, "Pretrained language model in continual learning: A comparative study," in *Proc. Int. Conf. Learn. Representations*, 2022.



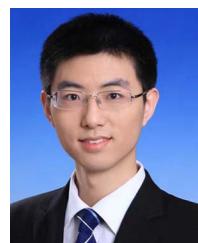
**Shirui Pan** (Senior Member, IEEE) received the PhD degree in computer science from the University of Technology Sydney (UTS), Ultimo, NSW, Australia. He is a professor with the School of Information and Communication Technology, Griffith University, Australia. Prior to this, he was a senior lecturer with the Faculty of IT at Monash University. His research interests include data mining and machine learning. To date, he has published more than 100 research papers in top-tier journals and conferences, including *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Transactions on Knowledge and Data Engineering*, *IEEE Transactions on Neural Networks and Learning Systems*, ICML, NeurIPS, and KDD. His research has attracted more than 20 000 citations. His research received the 2024 CIS IEEE TNNLS Outstanding Paper Award and the 2020 IEEE ICDM Best Student Paper Award. He is recognised as one of the AI 2000 AAAI/IJCAI Most Influential Scholars in Australia (2021). He is an ARC Future fellow and a fellow of the Queensland Academy of Arts and Sciences (FQA).



**Linhao Luo** received the bachelor degree from the Harbin Institute of Technology, Shenzhen, in 2021. He is currently working toward the PhD degree with the Faculty of Information and Technology, Monash University. His research interests include machine learning, data mining, and graph neural networks.



**Yufei Wang** received the bachelor's degree jointly from the University of Queensland and Sun-Yat Sen University, in 2016, and the master's and PhD degrees from Macquarie University, under supervision of Prof. Mark Johnson, in 2019 and 2023, respectively. He is a research associate with Monash University, Australia. His research interests include large language models, natural language processing, and controllable text generation.



**Chen Chen** received the bachelor's degree from the University of Science and Technology Beijing, China, in 2012, and the Msc degree from the University of New South Wales, Australia, in 2018. He is currently working toward the doctor of philosophy (PhD) degree with Nanyang Technological University, Singapore. His research interests include the area of natural language processing, knowledge graphs, and large language model.



**Jiapu Wang** is currently working toward the PhD degree with the Beijing Municipal Key Laboratory of Multimedia and Intelligent Software Technology, Beijing University of Technology, Beijing. His research interests include knowledge graph completion, computer vision, and pattern recognition.



**Xindong Wu** (Fellow, IEEE) received the bachelor's and master's degrees in computer science from the Hefei University of Technology, China, and the PhD degree in artificial intelligence from the University of Edinburgh, Britain. He is director and professor with the Key Laboratory of Knowledge Engineering with Big Data (the Ministry of Education of China), Hefei University of Technology, China. He is also a senior research scientist with Zhejiang Lab, China. His research interests include Big Data analytics, data mining, and knowledge engineering. He is a foreign member of the Russian Academy of Engineering, and a fellow of the AAAS (American Association for the Advancement of Science). He is the Steering Committee Chair of the IEEE International Conference on Data Mining (ICDM), and the editor in-chief of Knowledge and Information Systems (KAIS, by Springer). He was the editor-in-chief of *IEEE Transactions on Knowledge and Data Engineering* (TKDE) between 2005 and 2008, and co-editor-in-chief of the *ACM Transactions on Knowledge Discovery from Data Engineering* between 2017 and 2020. He served as a program committee chair/co-chair for ICDM 2003 (the 3rd IEEE International Conference on Data Mining), KDD 2007 (the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining), CIKM 2010 (the 19th ACM Conference on Information and Knowledge Management), and ICBK 2017 (the 8th IEEE International Conference on Big Knowledge). One of his completed projects is Knowledge Engineering With Big Data (BigKE), which was a 54-month, 45-million RMB, 15-institution national grand project, as described in detail at <https://ieeexplore.ieee.org/abstract/document/7948800>