Discover Artificial Intelligence

Review

A survey on augmenting knowledge graphs (KGs) with large language models (LLMs): models, evaluation metrics, benchmarks, and challenges

Nourhan Ibrahim[1,2] · **Samar Aboulela**[1] · **Ahmed Ibrahim**[3] · **Rasha Kashef**[1]

Received: 3 September 2024 / Accepted: 10 October 2024
Published online: 04 November 2024

Abstract
Integrating Large Language Models (LLMs) with Knowledge Graphs (KGs) enhances the interpretability and performance of AI systems. This research comprehensively analyzes this integration, classifying approaches into three fundamental paradigms: KG-augmented LLMs, LLM-augmented KGs, and synergized frameworks. The evaluation examines each paradigm's methodology, strengths, drawbacks, and practical applications in real-life scenarios. The findings highlight the substantial impact of these integrations in fundamentally improving real-time data analysis, efficient decision-making, and promoting innovation across various domains. In this paper, we also describe essential evaluation metrics and benchmarks for assessing the performance of these integrations, addressing challenges like scalability and computational overhead, and providing potential solutions. This comprehensive analysis underscores the profound impact of these integrations on improving real-time data analysis, enhancing decision-making efficiency, and fostering innovation across various domains.

Keywords Large language models (LLMs) · Knowledge graphs (KGs) · Retrieval augmentation generation (RAG) · Deep learning (DL) · Evaluation metrics

1 Introduction

The ubiquitous integration of data across industries such as healthcare, finance, telecommunications, and e-commerce has made the ability to manage and analyze large, interconnected datasets increasingly critical. Traditional data management systems often fall short when handling the complexity and scale of modern datasets, leading to inefficiencies and challenges in informational retrieval, recommendation systems, and real-time decision-making [1]. For instance, when data is fragmented across multiple sources, it becomes difficult to get a comprehensive view, resulting in incomplete information and sub-optimal decision-making. Incomplete data or disconnected systems can hinder the ability to spot patterns or anomalies, leading to missed opportunities or errors. In addition, data silos, where information is isolated within different departments or systems, can impede seamless access and integration. This fragmentation can limit the ability to personalize services, optimize operations, and improve overall efficiency. These issues highlight the need for more advanced tools and technologies that offer structured data management and enhanced semantic understanding

✉ Nourhan Ibrahim, nourhan.ibrahim@torontomu.ca; Samar Aboulela, samar.g.aboulela@torontomu.ca; Ahmed Ibrahim, aibrah64@uwo.ca; Rasha Kashef, rkashef@torontomu.ca | [1]Electrical, Computer, and Biomedical Engineering, Toronto Metropolitan University, Toronto, Ontario, Canada. [2]Faculty of Engineering, Alexandria University, Alexandria, Egypt. [3]Computer Science, Western University, London, Ontario, Canada.

[2]. Knowledge Graphs (KGs) [3, 4] and Large Language Models (LLMs) [5, 6] provide a more holistic view of data, improve integration, and enable more accurate and efficient decision-making.

1.1 Overview of knowledge graphs and LLMs

Knowledge Graphs (KGs) have recently emerged as one of the most compelling techniques for organizing and querying large, complexly interlinked datasets. They provide semantic understanding and reasoning, successfully applied in various domains, including search, recommendation systems, and data integration. KGs offer a semantic framework that supports complex data queries and reasoning [7]. They consist of nodes representing entities or concepts, edges showing relationships between them, and properties adding features to nodes and edges. One of the most significant strengths of KGs is their ability to narrow queries to small data subsets, maintaining query performance irrespective of dataset growth. This contrasts with relational databases and NoSQL systems, where query performance degrades as the size of the dataset increases. KGs can quickly fetch relevant data across relationships via high-speed graph traversal algorithms, making them essential for real-time data analysis and decision-making [8, 9].

Large language models are deep learning models trained with large text corpora, conditioned to understand the context and generate responses like humans. Some prevalent cases are GPT-3 by OpenAI, touted to create coherent, relevant text, and BERT by Google, which focuses on understanding words in their context for many NLP tasks. LLMs have changed a lot in the field of NLP, attaining milestones in text generation, machine translation, sentiment analysis, and conversation AI. They are characterized by the processing and generation of contextually aware and semantically rich. One of the most compelling features of LLMs is their complete flexibility and transferability across various domains. They can also be fine-tuned on specific tasks to guarantee high accuracy and performance of specialized applications. With millions to billions of parameters, they can master fine-grained language patterns and subtle concepts during training. At this heavy parameterization, they can reason out complex problems and produce summaries and question-answering tasks with very high precision. The advanced architecture of LLMs, which includes attention and transformers that let them figure out which words in a sentence are the most important, gives them the strength they need to handle a wide range of NLP tasks. This helps them capture context better than traditional models. Therefore, LLMs generate grammatically correct, contextually appropriate, and coherent texts [5, 6].

1.2 Research methodologies and contributions

This survey paper extensively reviews KGs, LLMs and their integration, focusing on how these advanced technologies can effectively enhance artificial intelligence systems. By examining various integration paradigms, the paper provides insights into each approach's methodologies, advantages, and limitations [5]. We adopted a multi-phase methodology to systematically analyze the integration of Knowledge Graphs (KGs) and Large Language Models (LLMs). Each phase was designed to comprehensively explore existing techniques, evaluate challenges, and propose future directions for research. Below, we outline the research methodology adopted in our survey paper as follows:

- We defined research objectives to explore how integrating KGs and LLMs enhances interpretability, performance, and applicability across NLP tasks.
- We conducted a systematic literature review, gathering key papers from NLP, machine learning, and knowledge representation to understand integration approaches in the last decade.
- We categorized the integration approaches into three paradigms, KG-augmented LLMs, LLM-augmented KGs, and synergized frameworks, to organize and streamline the analysis.
- We designed a comparative framework to analyze integration approaches based on key factors like accuracy, computational efficiency, scalability, and generalization capabilities.
- We identified commonly used datasets and benchmarks from the literature to evaluate and compare the performance of various integration techniques.
- We selected evaluation metrics, including quantitative measures (accuracy, precision, recall) and qualitative aspects (interpretability, relevance), to assess the effectiveness of KG and LLM integration.
- We highlighted key challenges in integrating KGs and LLMs, such as scalability, data privacy, and the need to maintain updated KGs for accurate performance.

- We explored real-world applications by providing case studies where KGs and LLMs are successfully integrated, such as in enhancing search engines and developing personalized dialogue systems.
- We proposed future research directions, including developing efficient integration techniques, enhancing real-time learning, and mitigating biases in LLMs using KGs.

In contrast to previous surveys that focus primarily on either Knowledge Graphs (KGs) or Large Language Models (LLMs) in isolation, this survey provides a comprehensive review of the integration of KGs and LLMs. The following are the key elements of uniqueness that differentiate this survey from existing literature:

- The survey explores the synergistic potential of integrating KGs and LLMs across three key paradigms: KG-augmented LLMs, LLM-augmented KGs, and Hybrid synergized frameworks
- Unlike earlier works, which often examine these technologies independently or focus narrowly on specific applications (e.g., semantic search or question-answering systems), our survey adopts a holistic perspective by Covering a broader spectrum of integration techniques and underlying architectures and highlighting their impact on improving AI systems' interpretability, performance, and reasoning capabilities.
- The survey identifies specific challenges arising from integrating KGs and LLMs, which are not sufficiently addressed in previous literature, including Data privacy concerns, Maintenance of up-to-date knowledge bases, and Computational overhead.
- By discussing the advancements and obstacles in the field, this survey provides a structured framework for understanding the interactions between KGs and LLMs. It paves the way for future research directions and practical applications that benefit from their combined use.

The following contributions highlight the key findings and implications of this survey paper:

- Presenting the three main integration paradigms-KG-Augmented LLMs, LLMs-Augmented KGs, and Synergized Frameworks-including how they work and their pros and cons.
- Exploring methodological insights into how Knowledge Graphs (KGs) and Large Language Models (LLMs) are integrated, including the techniques and approaches used to combine structured and unstructured data.
- Evaluating the advantages and disadvantages of each integration paradigm, such as improvements in text accuracy and context for KG-Augmented LLMs, enhancements in KG quality and functionality for LLMs-Augmented KGs, and potential biases and consistency issues.
- Discussing practical applications across various domains, including healthcare, finance, and e-commerce, and illustrating how these paradigms improve search engines, recommendation systems, and decision-making processes.
- Reviewing evaluation metrics and benchmarks essential for gauging the integration performance between Large Language Models (LLMs) and Knowledge Graphs (KGs) and illustrating how these standards help enhance accuracy, efficiency, and functionality in various computational tasks and applications.
- Addressing challenges related to integrating LLMs with KGs, such as scalability, computational overhead, and alignment between structured and unstructured data, and discussing potential solutions and strategies.

Integrating LLMs with KGs enhances their capabilities by automatically extracting structured information from unstructured texts, thereby improving the construction and enhancement of KGs [5, 6]. LLMs can detect and correct errors, add semantic depth, and provide contextual enrichment, leading to more accurate and coherent KGs. Additionally, LLMs can transform natural language queries into formal queries, making KGs more accessible and usable to a broader audience. This integration fosters advanced healthcare, finance, and commerce applications, where real-time data analysis and decision-making are crucial [10]. Together, KGs and LLMs create a robust framework for managing and analyzing complex data, aiding companies and researchers in developing intelligent systems that deliver accurate and precise contextual information across various domains at the right time. KG-augmented LLMs, which integrates knowledge graphs to enhance LLMs performance and interpretability; LLMs-augmented KG, whereby LLMs improve the quality and functionality of Knowledge Graphs; and Synergized LLMs + KG, which refers to mutual integration into one framework [5, 6]. The rest of this paper can be structured as follows: Sect. 2 provides background on LLMs, while Sect. 3 presents a detailed discussion of knowledge graphs. Section 4 introduces the various LLMs and KGs integration paradigms and their methodologies. Section 5 presents multiple case studies in advanced applications of combined LLMs and KGs. Section 6 discusses the expected benefits of combining both LLMS and KGs. Various evaluation metrics and benchmarks

are presented in Sect. 7. Section 8 discusses challenges and limitations, and finally, Sect. 9 concludes the paper with future research directions.

2 Background on large language models (LLMs)

This section briefly discusses the background of large language models (LLMs) and Knowledge Graphs (KGs).

2.1 Fundamentals of LLMs

Large language models (LLMs) are state-of-the-art AI models thoroughly pre-trained on massive amounts of text. These models have yielded impressive results across different Natural Language Understanding (NLU) and Natural Language Generation (NLG) tasks. Artificial intelligence and natural language processing advances are related to the development of LLMs. For instance, statistical models like n-grams and Hidden Markov Models (HMMs) were created in the 1990s, constituting an important milestone in language modeling. Similarly, with the emergence of word embedding methods like Word2Vec [11] and GloVe [12] that resulted in increasingly complex structures, the next decade witnessed the dominance of neural networks and deep learning. However, Vaswani et al. [13] achieved a major milestone in 2017 when they introduced transformer models upon which modern LLMs such as BERT [6] and GPT [5] were built. These transformer models use a self-attention mechanism that helps them process texts more efficiently and accurately. This has dramatically improved how language models work, making them, through these processes, very proficient in multifaceted linguistic activities. Some examples of remarkable LLMs are OpenAI's GPT series [14], Google's BERT [6], T5 [15], PaLM [16] and Gemini [17], as well as Meta's RoBERTa [18], OPT [19]and LLaMA [20]. These models are state-of-the-art LLMs, and each has added something new to advance this field.

- *GPT Series by OpenAI* The Generative Pre-trained Transformer (GPT) series comprising GPT-2 [21], GPT-3 [5] and the latest GPT-4 [22] created standards for Natural Language Processing. It is known for its ability to generate high-quality text and perform tasks such as translation, question answering, summarization, or engaging in coherent dialogues. For example, GPT-3 with 175 billion parameters can generate good quality text and do a lot of other functions, like translating a given passage from one language to another or producing summaries after reading long articles.[5].
- *Google's Models* Google has created many powerful LLMs like BERT (Bidirectional Encoder Representations from Transformers) [6], T5 (Text-To-Text Transfer Transformer) [15], PaLM (Pathways Language Model) [16], Gemini [17] and LaMDA (Language Model for Dialogue Applications) [20]. BERT [6] made a big impact on the field by introducing bidirectional training, which allows the model to look at the context from both sides, thus improving its understanding of language. T5 has unified multiple NLP tasks into a text-to-text framework, showing versatility across various applications. PaLM is a large-scale model trained with a Pathways system that efficiently scales many tasks and languages to enhance generalization and performance. To push the boundaries of what LLMs are capable of, Google's most recent model, Gemini, aims to have multi-modal capabilities. LaMDA targets conversational AI specifically for dialogue in an attempt to generate responses that are more natural and meaningful.
- *Meta's Models* RoBERTa (Robustly optimized BERT approach) [18] is one of the models Meta has contributed with, alongside OPT (Open Pre-trained Transformer) and LLaMA (Large Language Model Meta AI). Different pre-training strategies were considered during the development of RoBERTa compared to BERT, leading to better optimization. However, training on more data over longer periods achieved this, resulting in stronger performance across different NLP benchmarks. Openness and reproducibility were put at the core focus points by the OPT model, which provides a solid base for developing and experimenting with LLMs while being able to repeat results later if necessary. The most recent LLM that Meta has released prioritizes efficiency and scalability to provide everyone with access to reliable language models.

Table 1 showcases the key features and attributes of various language models that OpenAI, Google, and Meta developed. The models compared include OpenAI's GPT-2, GPT-3, and GPT-4; Google's BERT, T5, PaLM, Gemini, and LaMDA; and Meta's RoBERTa, OPT, and LLaMA. Each organization brings unique capabilities to its models. OpenAI's models are known for high-quality text generation, translation, question answering, and summarization, with GPT-3 excelling in multiple functions due to its 175 billion parameters. Google's models each have distinct features: BERT introduced bidirectional training for better language understanding, T5 uses a versatile text-to-text framework, PaLM enhances performance

Table 1 Comparison of language models

	OpenAI (GPT Series)	Google's models	Meta's models
Models	GPT-2, GPT-3, GPT-4	BERT, T5, PaLM, Gemini, LaMDA	RoBERTa, OPT, LLaMA
Common Features	Transformer-based architecture, designed for diverse text generation tasks	Transformer-based models; focus on varied applications like bidirectional training (BERT), multi-task learning (T5), and conversational AI (LaMDA)	Transformer-based models with emphasis on openness, efficiency, and accessibility
Strengths	Strong contextual understanding, large-scale pre-training for a wide range of language tasks, high-quality text generation	BERT: improved context understanding with bidirectional learning, T5: unified text-to-text framework, PaLM: efficient scaling, Gemini: multi-modal learning, LaMDA: specialized for dialogue	RoBERTa: optimized BERT with longer training, OPT: focus on model reproducibility, LLaMA: designed for efficiency and resource scalability
Improvements	GPT-3 scaled to 175 billion parameters, increasing performance in multi-functional NLP tasks; progression to GPT-4 further enhanced capabilities	BERT set benchmarks for NLP understanding, T5 covers multiple NLP tasks effectively, PaLM optimized for scalability, LaMDA enhanced dialogue-based AI	RoBERTa refined training strategies, OPT offers transparent and reproducible research models, LLaMA emphasizes efficiency while maintaining high performance
Similarities	Use of transformer architecture and transfer learning for diverse NLP tasks	Transformer-based designs; shared emphasis on natural language understanding and adaptability across models	Focus on efficient, transparent transformer models; all designed for scalability and varied NLP tasks
Differences	Specializes in high-quality text generation and generalization across tasks	Emphasizes bidirectional understanding (BERT), multi-task learning (T5), and conversational AI (LaMDA)	Prioritizes efficiency (LLaMA), openness and reproducibility (OPT), and optimization (RoBERTa)

across tasks and languages with efficient scaling, Gemini aims for multi-modal capabilities, and LaMDA focuses on generating natural and meaningful dialogue. Meta's models prioritize optimization and efficiency: RoBERTa is an optimized version of BERT with better performance across NLP benchmarks, OPT emphasizes openness and reproducibility, and LLaMA focuses on providing reliable, scalable, and efficient language models. This table provides a concise overview of the advancements and distinctions each organization contributes to Natural Language Processing through their respective models. Figure 1 illustrates the key milestones in developing LLMs, showcasing the major advancements in the field.

2.2 Capabilities and applications of large language models

LLMs have revolutionized the natural language processing (NLP) field by enabling the completion of various tasks. They are particularly adept at generating text, showcasing their abilities in creative writing, dialogue systems, and content creation [5]. Text summarizing - distilling key information from long documents very fast - is equally good as translating text between different languages with high precision [15].

Another impressive functionality of LLMs is their question-answering capabilities. They give accurate answers depending on the context; this can be applied to virtual assistants or customer support where correct feedback must be given [5] [6]. Besides sentiment classification within texts, LLMs also categorize them into topics and do name entity recognition (NER), where entities like names, dates, or locations are identified and classified in the text [6]. Completing sentences by predicting the following words comes naturally to them while maintaining original meaning simultaneously; equally, so does rewriting of texts while retaining initial sense [21]). In addition to learned abilities, some emergent capabilities come from extensive training but are not explicitly programmed into LLMs themselves, making them even more useful: zero-shot learning - models perform tasks without examples by understanding instructions expressed in ordinary language; few-shot learning - models solve new tasks effectively with just a few examples [5]. Common sense reasoning is also done when applying general world knowledge to conclude, whereas multi-task learning allows different kinds of tasks to be addressed within one model [5, 16]. Maintaining context over long texts helps give coherent responses in dialogue systems, while the ability to generate and understand code is known in Codex-like [24] models that assist software development tasks. LLMs also have abstract and analytical reasoning abilities to generate hypotheses, write scientific abstracts, or do basic arithmetic and logical operations [5]. Examples include GPT-3, which can perform a wide range of NLP tasks without prior training using natural language instructions [5]; BERT, with deep contextual understanding skills for question answering or NER task completion [6]; T5, which uses a unified text-to-text approach towards various language-related objectives [15]. PaLM has shown strong reasoning and language understanding capabilities [16], while in code-related tasks, Codex can be specifically helpful [24], thereby demonstrating the diversity of applications available for LLMs. With more research being conducted and larger/more diverse datasets used during training phases, these models will continue expanding their abilities and having a more significant impact across different AI domains.

2.3 Large models: size, type, and architectures

Language Models (LMs) can be classified in size, type, and availability [25]. In terms of size, small LMs are models that have one billion or fewer parameters, such as LLaMA-1. Medium LMs contain between one billion and ten billion parameters, such as GPT-2 and BERT. LLMs refer to models with ten billion to a hundred billion parameters like GPT-3 and PaLM. Very large language models include those ranging from a hundred billion to one trillion parameters, such as GPT-4. Regarding the type, foundation models are trained without specific instructions for their use cases, e.g., GPT-3, but instruction models are pre-trained and then fine-tuned according to specific tasks like T5. In contrast, chat models are pre-trained and then developed explicitly for chatting purposes only, e.g., ChatGPT. Availability is shown by the distinction between private (e.g., GPT-4) and public (e.g., LLAMA) models [25]. Different architectures (Table 2)- based on transformer models [13] - are used to build LLMs and determine their text processing and generating processes. For example, in the encoder-only architecture, which BERT represents [6], its focus is on understanding and representing text whereby it processes the whole input sequence at once, thus capturing bidirectional context hence being useful for tasks such as entity recognition or text classification.

On the other hand, GPT models are designed with a decoder-only architecture that can be used for language generation. Here, input is processed sequentially, and prediction about the next word depends on previous words; therefore, it's suitable for tasks like language modeling or text completion. This structure generates cohesive and

Fig. 1 Historical Development of LLMs

relevant texts within context most of the time [14]. Encoder-decoder architectures combine both strengths of encoder and decoder as seen in T5 [15]or BART (Bidirectional and Auto-Regressive Transformers) [23]. The input sequence is processed by an encoder, which creates a representation rich with context that is later used by the decoder to generate the output sequence. Such an architecture can perform many NLP tasks, including but not limited to translation, summarization and question answering, among others, since it's highly flexible [15].

Table 2 Transformer Model Architectures

Architecture type	Representative models	Description and applications
Encoder-only	BERT [6]	Focuses on understanding and representing text by processing the whole input sequence at once, capturing bidirectional context. Useful for tasks like entity recognition or text classification.
Decoder-only	GPT [14]	Designed for language generation. Processes input sequentially and predict the next word based on previous words. Suitable for tasks like language modeling or text completion.
Encoder-decoder	T5 [15], BART [23]	Combines strengths of encoder and decoder. The encoder processes the input sequence to create a rich contextual representation, which the decoder uses to generate the output sequence. Highly flexible, performing tasks such as translation, summarization, and question answering.

2.4 Challenges and limitations of LLMs

LLMs have made significant improvements in natural language understanding and generation. However, these models face several limitations that can be mitigated by integrating knowledge graphs. Firstly, LLMs rely heavily on information from the internet, which in most cases is vast but incomplete and often untrue. Such models are trained on massive datasets that contain accurate and inaccurate information. Consequently, they may generate content that propagates misconceptions. For example, they could wrongly state scientific facts because they do not inherently verify truthfulness while processing historical events. To tackle this problem, knowledge graphs should provide verified databases with current records about things [25]. Another major challenge is the need for deep contextual understanding exhibited by LLMs. However, these systems excel at producing human-like sentences; they fail to comprehend complex queries fully, especially those necessitating multi-step reasoning or significant background knowledge. This limitation becomes noticeable in applications needing appreciation of specific contexts at a finer grain level. By linking related entities and concepts in a structured way, knowledge graphs foster better context awareness among LLMs, which can then retrieve relevant information quickly enough while also relating different pieces correctly, leading to more precise response provision within appropriate situational settings [26].

LLMs may need specialized domain knowledge, even after training on diverse datasets. This is particularly evident in medical fields where precise and detailed information is crucial. Therefore, ensuring accuracy when dealing with any information related to these domains is essential. For instance, LLMs may provide basic guidance but need more ability to provide precise and dependable suggestions for specific medical issues. Knowledge graphs explicitly designed for these sectors will provide comprehensive information, allowing the system to consistently generate precise outputs whenever appropriate. One way to ensure correct diagnoses and treatment options by LLMs is by integrating a medical knowledge graph [27]. Moreover, LLMs must improve at making inferences, especially when dealing with complex queries involving many entities and relationships. Knowledge graphs represent connections well and are good at logical thinking; thus, they can perform logical inferences logically. Given this fact, it would be better if structured data from knowledge graphs were utilized more effectively during inferencing processes by LLMs rather than leaving everything solely upon these structures alone without further intervention from them [5]. A detailed discussion on knowledge graphs is provided next.

3 Knowledge graphs (KGs)

Knowledge graphs offer structured, verified, and contextually rich knowledge, improving LLMs' precision, contextual comprehension, domain-specific knowledge, and inferencing abilities. When these two components are combined, they create a powerful synergy that can result in more accurate AI systems that can handle complex and specialized queries, thus ultimately enhancing the performance and trustworthiness of LLMs in different application areas.

3.1 Fundamentals of knowledge graphs

Knowledge Graphs (KGs) are structured knowledge representations that link entities and their relations in a graph. This allows for complex information integration and querying across different domains. For instance, the Google Knowledge Graph [28] integrates search results with linked data to provide users with more contextually relevant and comprehensive information about entities directly on the results page. DBpedia [29] is another example where structured content from Wikipedia is extracted and made available online as a community-driven project. Such knowledge graphs help to organize and retrieve data, thus making it more usable in various applications [8].

Knowledge Graphs can be traced back to Tim Berners-Lee's Semantic Web vision of creating a machine-understandable web of data [36]. The term "Knowledge Graph" gained popularity after Google launched its version in 2012 [28]; this move combined linked open data with search results, thereby providing broader contexts along with richer details about searched items at once. Over time, other initiatives like DBpedia [29], Wikidata [34], etc., have broadened KGs' scope and usefulness within modern systems of knowledge organization [35]. Knowledge Graphs (KGs) are composed of foundational elements that structure and represent information meaningfully. The core components of a KG include nodes and edges, where nodes represent entities such as people, places, organizations, concepts, or events, and edges denote the relationships between these entities. These nodes and edges form the backbone of any graph-based representation, enabling a structured representation of facts and relationships within a domain. However, the value of a KG extends

beyond these structural elements; it also requires an ontology - a schema or structure that defines the types of entities, relationships, and associations within the domain context. This ontology provides semantic context and allows the KG to support reasoning and knowledge inference effectively [33]. Figure 2 illustrates a sample KG that depicts the relationships between various academic research entities, including research papers, authors, institutions, and fields of study.

Additionally, KGs often feature properties or attributes that provide extra facts about entities such as dates, locations, numerical values, etc., while inference mechanisms built into them allow for deducting new information from already known ones through logical reasoning; moreover, scalability is also an important aspect where knowledge graphs should grow easily with time by absorbing additional datasets without breaking themselves apart nor losing any interconnections among various parts. This ability to scale up while combining data from many sources makes KGs very effective tools for managing complex information systems [33].

3.2 Types of knowledge graphs

Table 3 shows the categories of Knowledge Graphs (KGs), which can be distinguished based on their structure, intended use, and coverage. Domain-specific KGs are focused on specific knowledge areas such as healthcare, finance, supply chain, and entertainment, among others, and contain highly specialized and detailed information in those fields. Many examples include SNOMED CT [30], which is a set of clinical terminologies; FIBO [31], which is a set of financial concepts; and SupplyChainKG [32], which combines information from suppliers, manufacturers, and logistics providers to make supply chains run more smoothly and safely.

Cross-domain KGs like Google Knowledge Graph [28] or DBpedia [29] cover broad areas by integrating knowledge from different disciplines to provide a wide-ranging source of information about various subjects, typically utilized in applications that require deep general understanding like search engines or virtual assistants [8].

For an organization, there may be a need for its knowledge representation system designed within its boundaries called an Enterprise KG; this captures internal data about processes carried out within it together with relationships among various entities, thus fostering business intelligence improvement as well as decision-making support systems [33].

The last category is called open KGs, e.g., Wikidata [34], these can be accessed by anyone worldwide and allow users to contribute towards its growth through continuous expansion as well refinement of knowledge base [35]. Each of these different types of KGs contributes towards making data more accessible, improving information retrieval capabilities, and supporting complex queries across diverse domains and applications.

3.3 Business uses cases for Knowledge graphs

Knowledge Graphs are utilized in different areas to increase information retrieval and management efficiency and effectiveness. Below are some of the key business use cases where KGs have made significant impacts.

- *Information Retrieval and Search Engines* Search engines are one of the most prominent examples of how Knowledge Graphs are utilized to improve information retrieval. Traditional search engines rely heavily on keyword matching, often leading to irrelevant or incomplete search results. Knowledge Graphs, however, revolutionize this process by understanding the context and relationships between entities in a query. Knowledge graphs offer a more semantic understanding of queries and significantly enhance the user experience by delivering more accurate and contextually relevant search results [37].
- *Recommendation Systems (RSs)* RSs have become a cornerstone of digital businesses, from e-commerce platforms to streaming services. Traditional recommendation engines often rely on collaborative or content-based filtering, which can be limited in scope and may not fully capture the complex relationships between different entities (e.g., users, products, movies). Knowledge Graphs improve recommendation systems by mapping out the relationships between entities, allowing for more sophisticated and accurate recommendations. For example, in an e-commerce setting, a KG can link products based on user behavior and through semantic relationships such as brand, category, complementary products, and user reviews [38].
- *Clinical decision-making* Knowledge Graphs offer significant potential in the healthcare industry to improve patient care, clinical decision-making, and personalized medicine. Healthcare data is often siloed across various systems and formats, making it challenging to integrate and analyze comprehensively. Knowledge Graphs help address

Table 3 Knowledge Graphs Categories

Type of KG	Examples	Description and applications
Domain-specific	SNOMED CT [30], FIBO [31], SupplyChainKG [32]	Focused on specific knowledge areas such as healthcare, finance, supply chain, and entertainment. Contains highly specialized and detailed information. Useful for specific domain applications.
Cross-domain	Google Knowledge Graph [28], DBpedia [29]	Covers broad areas by integrating knowledge from different disciplines. Provides a wide-ranging source of information about various subjects. Typically used in search engines and virtual assistants.
Enterprise	Internal company-specific KGs [33]	Designed within an organization's boundaries. Captures internal data about processes and relationships among various entities. Supports business intelligence and decision-making systems.
Open	Wikidata [34], Linked Open Data [35]	Accessible by anyone worldwide. It allows users to contribute to its growth through continuous expansion and refinement. It enhances data accessibility and information retrieval and supports complex queries.

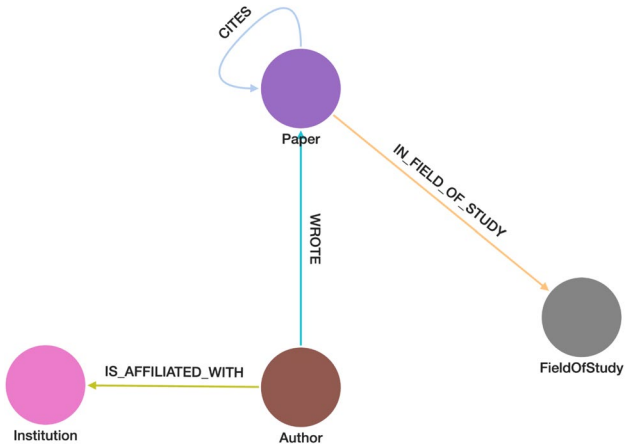

Fig. 2 A Sample KG for Academic Research Database

this issue by providing a unified, interconnected view of patient data, medical knowledge, and treatment protocols. One practical application is clinical decision support systems (CDSS), where KGs integrate patient data with vast medical literature, clinical guidelines, and drug information. By linking these diverse data sources, KGs enable healthcare providers to make more informed decisions. Furthermore, KGs play a crucial role in advancing personalized medicine. By linking genetic data with clinical records and medical literature, KGs can help identify potential disease biomarkers, suggest personalized treatment plans based on a patient's genetic profile, and predict potential adverse drug reactions [39].

- *Supply Chain Management (SCM)* Knowledge Graphs have proven invaluable in SCM. Supply chains are inherently complex, involving numerous entities such as suppliers, manufacturers, distributors, and customers. Managing these relationships and ensuring smooth operations require a comprehensive understanding of their interconnectedness. Knowledge Graphs can model the entire supply chain network, capturing the relationships between various stakeholders, products, locations, and transportation modes. This holistic view allows businesses to optimize their supply chain by identifying potential bottlenecks, optimizing inventory levels, and improving supplier relationships [40].
- *Fraud Detection and Prevention* Fraud detection is critical for many industries, particularly finance and insurance. Traditional fraud detection systems often rely on rule-based approaches that may miss sophisticated fraud schemes involving multiple entities and transactions. Knowledge Graphs offer a more effective solution by enabling the detection of complex patterns and relationships that might indicate fraudulent activity. A KG can model relationships between bank accounts, transactions, and individuals in the financial industry. By analyzing these relationships, the system can identify unusual patterns, such as a sudden increase in transactions between accounts with no previous connection, which might indicate money laundering or other fraudulent activities. Similarly, in the insurance industry, KGs can be used to detect fraudulent claims by linking policyholders, claims, medical records, and other relevant entities. If a KG identifies that a particular medical provider is connected to a disproportionately high number of similar injury claims, this could be flagged for further investigation. Knowledge Graphs can uncover hidden connections and patterns, making them a powerful tool in the fight against fraud [41].
- *Customer Relationship Management (CRM)* Customer Relationship Management systems are essential for businesses to manage interactions with current and potential customers. Knowledge Graphs enhance CRM by providing a more comprehensive and connected view of customer data, which can be used to improve customer service, marketing strategies, and sales processes. A knowledge graph can link a customer's purchase history, service interactions, social media activity, and feedback into a unified view. This holistic view allows businesses to understand their customers better and tailor their interactions accordingly. If a customer frequently purchases eco-friendly products, the CRM system can recommend related products, provide personalized marketing offers, and ensure customer service representatives know the customer's preferences. Additionally, KGs can help businesses identify potential upsell or cross-sell opportunities by analyzing relationships between products and customer segments. This capability leads to more targeted and effective marketing campaigns, ultimately driving higher customer satisfaction and loyalty [42].

3.4 KGs: challenges and limitations

Nevertheless, there are some limitations to Knowledge Graphs. Constructing and maintaining KGs may require significant resources as they involve much work on data integration, cleaning, and updating, among others. When the input is incomplete or incorrect, it can result in wrong conclusions being drawn, leading to unreliable outcomes. Besides this point, there might be difficulties in representing complex or subtle information that does not fit neatly into pre-defined schema using KGs alone. Also, privacy issues arise when sensitive data is included in knowledge graphs, necessitating strong safeguards for confidentiality, integrity, and security privacy protection [43].

4 Integration approaches and techniques

The integration of Large Language Models (LLMs) with Knowledge Graphs (KGs) has recently gained significant prominence due to these two technologies' complementary strengths. LLMs excel in natural language understanding and generation, whereas KGs provide structured and explicit knowledge, enhancing the performance and interpretability of LLMs. Three primary paradigms for integrating these models are KG-enhanced LLM, LLM-augmented KG, and Synergized LLMs + KG [44]. KG-enhanced LLMs focus on enhancing LLM performance and interpretability using KGs, while LLM-augmented KGs aim to improve KG-related tasks with the help of LLMs. The synergized framework integrates both technologies to enhance their capabilities, benefiting knowledge representation and reasoning in various applications [45, 46].

4.1 KG-Enhanced LLMs

This integration involves embedding KG into an LLM to improve performance and address issues such as hallucination or lack of interpretability. This involves representing entities and relations from a KG in continuous space vectors that an LLM can directly utilize at training or inference time. This can occur during pre-training, where an LLM is exposed to the knowledge within a KG, or during inference, where an LLM retrieves information from a KG to answer domain-specific queries. This integration offers structured ways to understand and trace the predictions made by the system [44] and [47]. KG-enhanced LLMs are further categorized into pre-training, inference, and interpretability. Pre-training methods incorporate KGs during the LLM training phase to enhance knowledge expression. In contrast, inference methods utilize KGs during the LLM inference phase to access the latest knowledge without retraining. Interpretability research uses KGs to understand the knowledge learned by LLMs and to interpret their reasoning processes [48, 49].

Models like KEPLER and Pretrain-KGE use BERT-like LLMs to encode textual descriptions of entities and relationships into vector representations. They are then fine-tuned on different KG-related tasks to make them work better [45]. Fine-tuning large language models (LLMs) means adapting pre-trained LLMs to effectively use structured information contained in KGs to generate contextually accurate responses. The extensive interlinked data in KGs makes this possible. The process begins with data preparation, extracting entities and relationships from KGs using techniques like Named Entity Recognition (NER) and relation extraction. The extracted data is embedded into continuous vector spaces by methods like node2vec, Graph Neural Networks known as GNNs, enabling the LLM to incorporate structured knowledge during training and inference., or Graph Neural Networks, referred to hereinafter as GNNs, allowing the LLM to incorporate structured knowledge during training and inference [50]. Figure 3 shows KG- Enhanced LLMs framework and outlines how KGs can be harnessed during various phases of LLM development to augment their performance regarding the support offered to LLMs. The framework shows the possibility of building the models and an efficient internal representation of KGs, which improves the performance of information retrieval tasks advanced by the LLMs, their understanding of the context, and accuracy in execution.

To perform fine-tuning, the LLM is trained on this embedded graph data so that its general understanding of language matches the structured knowledge from the KG. Fine-tuning the graph data improves the contextual features of the LLM, increases reasoning capabilities, and reduces hallucinations through the grounding of outputs in verified knowledge. It builds a bridge to other application domains: healthcare, by enhancing diagnostic tools and personalized medicine with the integration of medical knowledge graphs [51]; finance, by improving risk assessment and fraud detection; and suggesting enhanced investment-driven by the imitation of financial knowledge graphs [52]; and e-commerce, by

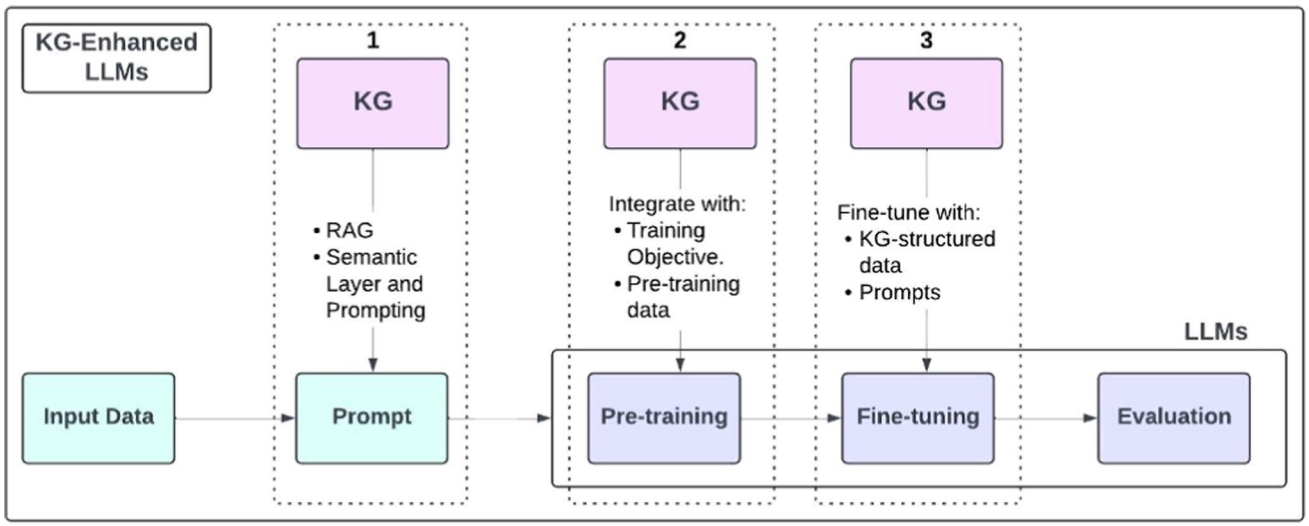

Fig. 3 KG-Enhanced LLMs Framework

enhancing recommendation engines and customer service, driven by the integration of product and customer knowledge graphs [53]. In other words, customizing LLMs on graph data works far more efficiently concerning complex, structured information. Hence, the output should be more accurate and contextually meaningful for different applications.

4.2 LLMs-augmented KG

This approach leverages the generalization capabilities of LLMs to perform tasks related to KGs more effectively. This includes processing text with an LLM to enrich graph representations, generating new facts by completing missing parts within a KG (known as "knowledge completion") and extracting entities and their relationships from texts to aid in constructing new graphs. Additionally, LLMs can generate human-like descriptions of facts in a KG, facilitating tasks such as KG-to-text generation and question-answering [43] and [47]. LLMs have the potential to significantly improve how KGS are built by processing large volumes of text data to extract relevant entities together with their relationships. The structure shown in Figure 4 shows how to combine large language models (LLMs) and knowledge graphs (KGs) to

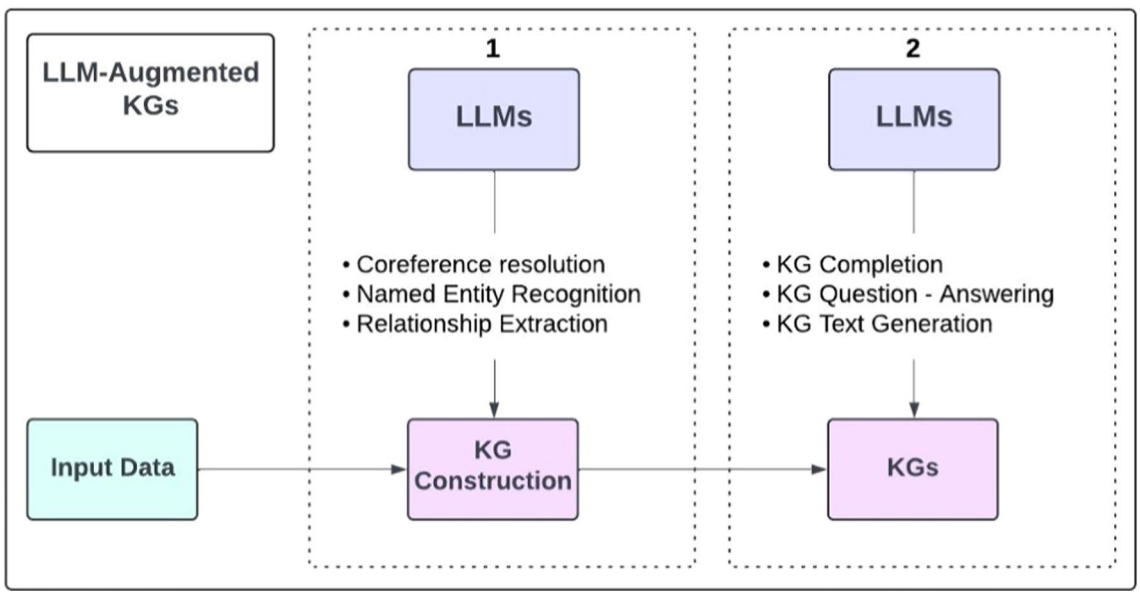

Fig. 4 LMMs Augmented KGs

design an LLM-augmented KGs process. The procedure is structured into two principal stages; the first involves applying LLMs to facilitate the synthesis of the KGs by performing tasks like coreference resolution, named entity recognition, and relationship extraction. These tasks enable end users to extract and relate to relevant entities and relationships from core aspects of the input document and thus develop a KG. In the second stage, other kinds of tasks are performed on the constructed KG with the help of LLMs; these tasks include KG completion, where LLMs are used to fill in the gaps of the attached graph, KG question answering, where LLMs are used to query the responses from the KG, and KG text generation, where descriptions of the nodes of the KG are developed using natural language. Such integration takes advantage of the features of both the LLMs, including successes in the generation and understanding of languages and KGs aspects of knowledge representation and structuring, making it possible to create a more effective and functional AI that can analyze complex data streams and make timely decisions.

Named entity recognition (NER), coreference resolution, relation extraction, etc., are techniques commonly applied to create detailed, accurate KGs. Another way could involve using language models to extract triples from unstructured texts, enriching them with new knowledge, which can then be added to the graph. This method uses the technology and improvement made possible by these models' language understanding skills when automatically building knowledge graphs [54] and [46]. LLMs can also be used to help manage databases, specifically graph databases. Complex queries in natural language can be translated into structured query language (SQL) or graph query language (GQL) using LLMs. This enables non-expert users to interact with the database more intuitively. In addition, LLMs can suggest relationships and entities for database schema design based on given data, thus improving the overall efficiency of database management systems. For instance, BERT and GPT-3 models have been employed to generate and optimize queries, providing a user-friendly interface for querying databases and enhancing query performance[55]. LLM-augmented KGs are categorized into five groups based on task types: KG embedding, KG completion, KG construction, KG-to-text generation, and KG question answering. These methods use LLMs to improve KG representations, encode text or make facts for KG completion, do tasks like entity discovery and relation extraction for KG construction, describe KG facts in natural language, and connect questions asked in natural language with answers based on KG [56–58].

4.3 Synergized LLMs + KG

This approach aims to create a unified framework in which LLMs and KGs mutually enhance each other's capabilities. This involves integrating multimodal data, employing techniques from both fields, and considering various real-world applications, such as search engines, recommendation systems, and AI assistants. Figure 5 depicts a general overview of a synergized framework that combines LLMs and KGs to enhance knowledge representation and reasoning. The LLMs and KGs in this framework work in a roundabout mechanism whereby LLMs use extensive structured knowledge from KGs to better their reasoning and understanding while KGs master language production and contexts of LLMs. This

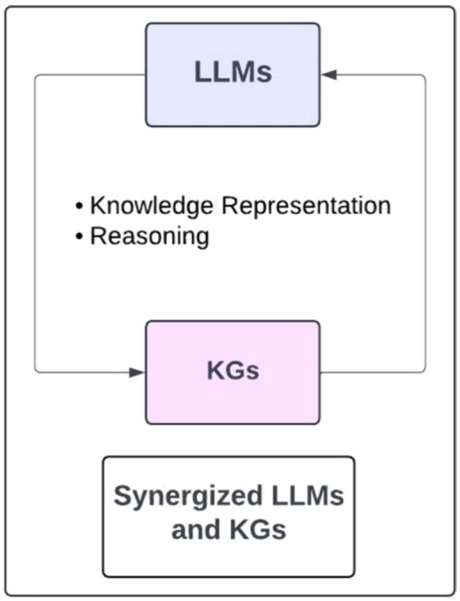

Fig. 5 LMMs Augmented KGs

mutual relationship between the two systems enables them to supplement each other effectively, giving rise to better and richer outputs. Such integration also allows models to answer complex queries, explain them more sophisticatedly, and provide verifiable information by drawing unstructured and structured data to supplement LLMs, for instance, using both KGs and LLMs. This improves the accuracy and understanding of AI systems, thus making such systems more useful when deployed in real life.

Combining LLMs with KGs can significantly improve knowledge representation and reasoning, making the overall system more robust and versatile [44] and [52]. This integration between LLMs and KGs will help in improving several future research directions, including using KGs for hallucination detection in LLMs, editing knowledge in LLMs, injecting knowledge into black-box LLMs, developing multi-modal LLMs for KGs, improving LLMs' understanding of KG structure, and enhancing bidirectional reasoning with synergized LLMs and KGs. [55, 59, 60].

4.4 Semantic layers and prompting techniques

Semantic layers are crucial integration dimensions between Large Language Models (LLMs) and Knowledge Graphs (KGs), acting as a bridge to map raw data into meaningful, interpretable forms that enhance the model's capabilities in understanding and generating text. They enable LLMs to draw more effectively on structured knowledge in KGs, resulting in enhanced output accuracy and contextual relevance.[43]. Essentially, they transform raw input into semantically rich representations, aiding LLMs in understanding and utilizing the underlying knowledge. Semantic parsing, entity linking, and relation extraction implement semantic layers. These techniques extract and infer critical concepts and relationships from the data, feeding them into the LLM during processing and response generation [43, 44]. The benefits of semantic layers include making LLMs more interpretable through structured context for outputs, which enhances the accuracy and contextual appropriateness of responses. This approach reduces instances of hallucination and improves the overall reliability of the model [50, 50].

Prompting techniques to control the behaviour of LLMs during text generation. These techniques involve designing specific inputs, called prompts, to guide the model's output to be relevant and contextually accurate. Prompting techniques vary, including direct prompts that provide explicit instructions or questions, contextual prompts that include background information to set the context for the model's understanding, and chained prompts, where a sequence of prompts is used to refine the model's responses incrementally [5, 52]. Effective prompts should be clear, concise, relevant to the task, and specific, with keywords and details instructing the model on generating an output. Applications of prompting techniques span question-answering, content generation, and interactive dialogue systems. They increase the relevance and usefulness of generated answers by guiding the model's response. However, designing effective prompts requires a deep understanding of the task and the model's behavior. This reliance on well-designed prompts can limit the model's flexibility in handling various inputs [61].

4.5 Industrial Use-cases

Doctor.ai is an example of a health care assistant combining LLMs and KGs to provide accurate medical advice through structured medical knowledge and natural language processing capabilities. Another instance is OpenBG, a recommendation systems-oriented knowledge graph that uses LLMs to process and understand user preferences from textual data, thereby improving recommendation accuracy [52]. **Neo4j** has integrated natural language processing tools for translating user queries into Cypher, its native graph query language. This enables users to utilize the power of graph databases without necessarily having deep technical expertise, thus significantly widening the accessibility and usability of graph database systems [62].

5 Case studies in advanced applications

5.1 Retrieval-augmented generation (RAG)

RAG involves combining LLMs with retrieval mechanisms to improve the quality and correctness of generated text. In this framework, a retriever model first retrieves relevant documents or passages from a large corpus given an input query. Subsequently, a generator model, usually an LLM, uses the retrieved information to generate coherent and contextually accurate responses. Thus, the responses are grounded on factual data by this method, thereby reducing

hallucination issues, among others, while increasing the relevancy of generated texts [60]. The RAG approach is beneficial for integrating KGs with LLMs. The retriever can query a KG to fetch relevant entities and relations, which are then used by the LLMs to produce informed, contextually rich responses. Such integration enhances the interpretability and factual consistency of LLMs outputs, thus making RAG suitable for question-answering systems and dialogue systems information retrieval, among other tasks [60]. RAG involves two steps, namely, information retrieval and text generation. The first step in RAG is to use external knowledge sources to obtain information related to the input query. Secondly, this information is utilized in the text-generation phase to produce contextually relevant responses. The extra information helps to make the generated text more accurate and precise by using reliable sources. Revealing the information sources also results in building user trust. In addition, using RAG can save money because it works with existing language models without extensive fine-tuning or retraining. RAG systems are not immune to hallucination, in which the text generated can have plausible-sounding but false information, requiring assurance mechanisms with regard to the content [60].

While the computational expense of RAG turns out to be considerable because it is a two-step process divided into retrieval and generation using vast computational resources, its paramount quality considerably depends on the quality of the retrieved data. Irrelevant or poor-quality data can end up giving wrong outputs. Scalability is another challenge due to the management and query of large data sets, where retrieval is significantly slowed, particularly when updates are frequently required [63]. Integration Complexity Even more, integration complexity is another reason why the synchronization of said retrieval and generation components increases the complexity of maintenance for RAG-based systems and thus may make it difficult for them to find wide use [64]. It is also difficult to evaluate RAG models because current metrics are not enough to measure the value that retrieval adds, and thus benchmarking proves to be difficult (compare to, e.g., [65].

5.2 Sequential fusion

By integrating information from complicated settings into domain-specific models, the work in [66] presents a novel Sequential Fusion technique that aims to improve LLMs. The suggested method is divided into two primary phases. First, KGs that extract structured knowledge from complex texts are built using general LLMs. In this phase, a relation extraction procedure is conducted under the direction of several prompt modules that offer detailed guidelines, intermediary reasoning processes, output formats, and advice on how to guarantee clarity and minimize ambiguity. The knowledge gathered in this phase is then organized into KGs, allowing for a complete grasp of the context. In the second step, a Structured Knowledge Transformation (SKT) module is used to convert the structured knowledge into descriptions in natural language. As a result of this transformation, the knowledge becomes more usable and appropriate for updating domain-specific LLMs. After that, the generated natural language descriptions are utilized to improve the LLMs through the Knowledge Editing (IKE) method, which incorporates the new knowledge without requiring significant retraining. This two-stage approach enables efficient updates, which improve the LLM's performance in specific tasks by incorporating correct and up-to-date information [66].

The Sequential Fusion approach [66] has many advantages. It tackles the difficulties of updating LLMs in situations requiring sophisticated reasoning with small sample sizes. The approach is helpful for instantly adapting new information since it can improve LLMs without requiring extensive retraining. Furthermore, the method guarantees that the updated LLMs retain high accuracy and comprehensibility by converting structured information into natural language, enhancing their performance in domain-focused tasks. This technique offers a solid and scalable answer to the problems associated with integrating and updating knowledge in large language models.

6 Benefits of integrating LLMs with KGs

Large Language Models (LLMs) are efficient tools for understanding and generating human language, but they usually find it difficult to access and verify factual information. Unlike them, Knowledge Graphs (KGs) preserve structured factual knowledge that can support LLMs by giving more data for interpretation and reasoning. Combining LLMs with KGs exploits both technologies' strengths, thus enhancing performance, knowledge extraction and enrichment, contextual reasoning and personalization, reliability, explainability, and scalability [67].

- Performance: When LLMs are integrated with KGs, natural language understanding and generation are greatly improved. Textual coherence that is contextually relevant is what LLMs excellently generate, but it may sometimes

lack the deep knowledge required by some queries. Therefore, the model can provide accurate responses when it accesses structured data within KGs while considering the context involved. For example, a particular historical event may need specific scientific or technical details that can be supplied by a KG, thereby improving the model's overall performance [26].

- Knowledge extraction and enrichment represent essential benefits of integrating LLMs with KGs. In this regard, KGs can use various sources to update and expand their databases continuously, thus broadening their scope of knowledge. Advanced language processing abilities possessed by LLMs may aid in identifying relevant pieces of information from unstructured texts during the extraction process so that they are included in the KG. This synergy guarantees currency and comprehensiveness of knowledge within KG, thereby enriching quality information available to LLMs [27].
- Contextual reasoning and personalization: LLMs have contextual-based language interpretation capabilities; however, they require assistance maintaining coherence over long conversations or grasping subtle points. Knowledge graphs provide a structured framework that connects related entities or concepts, helping LLMs track context during these interactions. This blending becomes more valuable when dealing with applications that need individualized responses, like customer service or adaptive learning systems. By using KG structured knowledge, LLMs can adapt their replies based on the personalized requirements of different users, thus giving them a personalized, relevant experience [68].
- Reliability is critical in AI models, especially in industries where wrong information can lead to serious consequences, such as healthcare, finance and legal services. Through the integration of LLMs together with KGs, models can cross-check the structured data against generated outputs, reducing the chances of errors or misinformation. Such a verification process ensures that results are founded on correct, valid data, allowing trustworthy AI systems [5].
- Explainability and transparency play an important role in adopting trustworthiness in AI technologies. LLMs serve as "black boxes" due to their complex nature, often leaving users wondering how they arrived at certain conclusions. Knowledge graphs have the potential to address this challenge by providing a clear, structured representation of what was known at each point in time used during reasoning paths taken by an artificial intelligent system. This will enable better interpretation, allowing people to trace back sources behind specific outputs, thereby enhancing explainability/transparency for AI systems [69].
- The scalability and efficiency of AI models can be improved by integrating LLMs with KGs. In a way that is easy to query and update, KGs can save large amounts of structured information, thus drastically reducing computational resources needed by LLMs to process massive datasets. This effectiveness is very important for real-time processing and response applications. By factoring out the storage and retrieval of factual knowledge to KGs, LLMs can concentrate on language generation and interpretation, resulting in quicker and more efficient AI systems [67].

7 Evaluation metrics and benchmarks

7.1 Evaluation metrics

Evaluating the performance of LLMs integrated with KGs is crucial to ensure their efficacy in various applications such as question answering, entity recognition, and semantic parsing. Several metrics are commonly used in the literature to assess these models, addressing different aspects of performance, from accuracy and relevance to execution and reasoning capabilities. Below is a comprehensive list of these metrics and their definitions and applications.

- *Accuracy* It measures the proportion of correctly predicted instances out of the total instances. This metric is widely used across tasks such as question answering, entity recognition, and text classification to gauge the accuracy of the model's predictions [70].

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

  Where TP, TN, FP, and FN are the true positives, true negatives, false positives, and false negatives, respectively.
- *ROUGE (Recall-Oriented Understudy for Gisting Evaluation)* It evaluates the quality of summaries by comparing the overlap with reference summaries using measures like precision, recall, and F1-score. This metric is commonly

used in summarization tasks to assess how well the generated summary captures the key information from the reference summary [70].

- *BLEU (Bilingual Evaluation Understudy)* It measures text quality by comparing it to reference texts. It is often used in machine translation and text generation tasks to evaluate how closely the generated text matches human-written references [70].

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{2}$$

Where BP (brevity penalty), w_n, and p_n are the weights and precision scores for n-grams.

- *Time Cost* It refers to the time taken to complete a task or process, indicating efficiency. This metric is used to assess the computational efficiency of the model, which is particularly important in real-time applications [71].
- *Comprehension* It assesses how well the model understands the graph structure and task. It is used in tasks that require a deep understanding of complex data structures like graphs, ensuring the model's comprehension capabilities [71].
- *Correctness* It evaluates the accuracy of the model's output against the expected results. This metric is used to verify that the model's outputs are not only accurate but also logically correct [71].
- *Fidelity* It measures how the model's reasoning process aligns with human logical reasoning. It is applied in reasoning tasks to ensure that the model's thought process is consistent with human logic [71].
- *Rectification Comprehension* It assesses the model's ability to correct its mistakes based on feedback or additional information. This metric is used to evaluate the adaptive learning capabilities of the model, particularly important in iterative learning environments [71].
- *Macro-F1* It is an average of the F1-scores calculated per class, treating all classes equally regardless of size. This metric is used in classification tasks to provide a balanced performance measure across different classes [70].

$$\text{Macro-F1} = \frac{1}{N}\sum_{i=1}^{N} F1_i \tag{3}$$

- *Training Time* It refers to the time to train the model on the dataset. This metric evaluates the efficiency of the training process, which is important for assessing the practicality of deploying the model [71].
- *Tuned Parameters* They denote the number of parameters adjusted during the tuning process to optimize model performance. This metric measures the complexity and effort of fine-tuning the model for optimal performance [71].
- *GPU Occupancy* It is the utilization rate of the GPU during model training and inference. This metric assesses the model's hardware efficiency and resource utilization [71].
- Mismatch Rate: It refers to the frequency of incorrect predictions or classifications. This metric identifies and quantifies the model's errors, essential for improving accuracy [71].

$$\text{Mismatch Rate} = \frac{M}{N_{\text{pred}}} \tag{4}$$

M is the number of mismatches and N_{pred} is the total predictions.

- **Denial Rate:** It is the rate at which the model fails to provide a valid prediction or answer. This metric measures the robustness of the model in handling various inputs without failing [71].

$$\text{Denial Rate} = \frac{D}{N_{\text{req}}} \tag{5}$$

Where D is the number of denials and N_{req} is the total requests.

- **Token Limit Fraction:** It is the proportion of input or output tokens that reach or exceed the model's token limit. This metric assesses the model's capability to handle long sequences without truncation issues [71].

$$\text{Token Limit Fraction} = \frac{T_{\text{lim}}}{T_{\text{tot}}} \tag{6}$$

Where T_{lim} and T_{tot} are the number of tokens reaching the limit and the total tokens processed, respectively.

- *Precision* It is the proportion of true positive predictions out of all positive predictions made by the model. This metric measures the accuracy of the model in identifying relevant instances correctly [70].

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{7}$$

- *Recall* It is the proportion of true positive predictions from all actual positive instances in the dataset. This metric measures the model's ability to identify all relevant instances in the dataset [70].

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{8}$$

- *F1-Score* It is a measure that combines precision and recall, calculated as the harmonic mean of precision and recall. It is used in binary and multi-class classification tasks to provide a balanced evaluation of the model's performance [70].

$$F1\text{-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{9}$$

- *Latency-Volume Trade-off* It evaluates the balance between the speed (latency) and the amount of data processed (volume). This metric is used to optimize the model for speed and capacity, which is important in large-scale data processing tasks [70].
- *Number of Errors and Cost* It refers to the total number of errors made by the model and the associated computational cost. This metric measures the model's accuracy and efficiency, providing a comprehensive performance evaluation [70].
- *Hits@k* It measures the proportion of times the correct answer appears in the top-k predictions. This metric is commonly used in retrieval tasks to evaluate the ranking performance of the model [72].

$$\text{Hits@k} = \frac{H_k}{N_{\text{query}}} \tag{10}$$

  Where H_k is the number of times the correct answer is in the top-k and N_{query} is the total queries.
- *Exact Match (EM)* It is the proportion of predictions that match the reference exactly. This metric is used in tasks requiring precise matching, such as closed-book question answering [72].

$$\text{Exact Match} = \frac{PEM}{N_{\text{pred}}} \tag{11}$$

  PEM is the proportion of exact matches and N_{pred} is the total predictions.
- *Mean Squared Error (MSE)* It measures the average of the squares of the errors between predicted and actual values. This metric is commonly used in regression tasks to quantify the difference between predicted and actual values [72].

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{12}$$

- *Human Evaluation* It is a subjective assessment by human evaluators, often used for generative tasks where objective metrics are insufficient. This metric provides qualitative insights into the model's performance, particularly in tasks involving language generation and comprehension [72].
- *Metrics Depending on Downstream Tasks* They vary depending on the specific downstream tasks and can include accuracy, F1-score, precision, recall, etc. These metrics are used to tailor the evaluation to the specific requirements of different applications [70, 71].

These evaluation metrics collectively provide a comprehensive framework for assessing the performance of LLMs integrated with KGs. This ensures that these models are accurate and efficient and deliver high-quality, contextually relevant, and logically sound outputs.

7.2 Benchmarks

Benchmarks are essential for evaluating the performance of Large Language Models (LLMs) integrated with Knowledge Graphs (KGs). They provide standardized datasets and evaluation metrics, allowing for consistent and comparative assessment of different models. Below are some of the key benchmarks used in this domain.

- *GLUE (General Language Understanding Evaluation)* It is a benchmark for evaluating the performance of models on a diverse range of natural language understanding tasks. It includes tasks such as sentence similarity, textual entailment, and sentiment analysis, providing a comprehensive measure of a model's understanding and reasoning capabilities [73].
- *SuperGLUE* It is an extension of GLUE, designed to be a more challenging benchmark for evaluating natural language understanding. It includes more difficult tasks such as causal reasoning and coreference resolution, providing a higher standard for evaluating model performance [74].
- *SQuAD (Stanford Question Answering Dataset)* It is a benchmark for evaluating question-answering systems, where models must read and answer questions based on that passage. It measures a model's ability to understand and retrieve information from text [75].
- *CommonsenseQA* It is a benchmark for evaluating commonsense reasoning in question-answering tasks. It focuses on the ability of models to use commonsense knowledge to answer questions, challenging their reasoning capabilities and understanding of everyday scenarios [76].
- *WikiKG90M* It is a large-scale benchmark for evaluating knowledge graph completion tasks. It includes tasks such as link prediction and entity classification, measuring a model's ability to understand and complete knowledge graphs [77].
- *Open Graph Benchmark (OGB)* It is a collection of realistic large-scale benchmark datasets for machine learning on graphs. It includes a variety of graph-based tasks such as node classification, link prediction, and graph classification, providing a comprehensive evaluation framework for models working with graph-structured data [78].
- *ATOMIC* It is an atlas of machine commonsense for evaluating models on commonsense reasoning. It provides a large-scale knowledge graph of everyday commonsense knowledge, used for tasks such as inference and explanation generation, making it a critical benchmark for assessing the ability of models to handle commonsense reasoning [79].
- *XNLI (Cross-lingual Natural Language Inference)* It is a benchmark for evaluating cross-lingual language understanding. It tests the ability of models to perform natural language inference across multiple languages, providing a measure of their multilingual comprehension and reasoning capabilities [80].
- *HellaSwag* It is a benchmark for evaluating commonsense reasoning in natural language. It tests models' ability to complete sentences coherently and sensibly, challenging them to demonstrate a deep understanding of context and everyday knowledge [81].
- *CLUE (Chinese Language Understanding Evaluation)* It is a benchmark for evaluating Chinese language understanding. It includes a variety of tasks such as sentence classification, reading comprehension, and machine translation, providing a comprehensive evaluation framework for models working with Chinese language data [82].
- *ReClor* It is a benchmark for evaluating logical reasoning in reading comprehension. It tests the ability of models to understand and reason through logical relationships in text, making it an important benchmark for assessing advanced comprehension and reasoning capabilities [83].
- *LAMA (LAnguage Model Analysis)* It is a benchmark for evaluating the factual knowledge contained in pre-trained language models. It tests the ability of models to recall factual information without additional context, providing a measure of their knowledge retention capabilities [84].
- *T-REx* It is a benchmark for evaluating the ability of models to understand and generate text based on structured data from knowledge bases. It tests the ability of models to generate coherent and factually accurate text from knowledge base triples, making it essential for assessing structured data processing capabilities [85].
- *NELL-995* It is a benchmark for evaluating knowledge extraction and completion from large-scale knowledge bases. It tests the ability of models to extract and infer new knowledge from existing knowledge base entries, providing a measure of their knowledge augmentation and completion capabilities [86].
- *WebQuestionsSP* It is a benchmark for evaluating question answering over knowledge graphs. It tests the ability of models to answer questions by querying structured data in knowledge graphs, providing a measure of their ability to work with structured query languages [87].

- *ComplexWebQuestions* It is a benchmark for evaluating complex question answering over knowledge graphs. It tests the ability of models to handle multi-hop reasoning and compositional questions over knowledge graphs, challenging their advanced reasoning capabilities [88].
- *MetaQA* It is a benchmark for evaluating multi-hop question answering over knowledge graphs. It tests the ability of models to perform multi-step reasoning over structured data, providing a measure of their complex query handling capabilities [89].
- *GrailQA* It is a benchmark for evaluating generalization in knowledge graph question answering. It tests the ability of models to generalize to unseen entities and relations in knowledge graphs, providing a measure of their robustness and adaptability [90].
- *SimpleQuestions* It is a benchmark for evaluating simple question answering over knowledge graphs. It tests the ability of models to answer straightforward, single-hop questions using knowledge graphs, providing a measure of their basic query handling capabilities [91].
- *FreebaseQA* It is a benchmark for evaluating question answering using the Freebase knowledge graph. It tests the ability of models to answer questions by querying the Freebase knowledge graph, providing a measure of their ability to handle large-scale structured data [92].

These benchmarks provide standardized datasets and evaluation metrics crucial for consistent and comparative assessment of LLMs integrated with KGs. They cover various tasks, including natural language understanding, question answering, commonsense reasoning, and knowledge graph completion, ensuring a comprehensive evaluation framework for these advanced models as shown in Table 4

8 Challenges and Limitations

8.1 Challenges related to language models

- *Hallucination* Sometimes, LLMs generate information that conflicts with existing sources (intrinsic hallucination) or cannot be verified (extrinsic hallucination). [44, 93].Alignment tuning and tool utilization can help alleviate the issue [93].
- *Knowledge Recency* LLMs' lack of access to the most recent data [93] could result in erroneous or outdated findings.
- *Implicit Knowledge* LLMs inherently retain knowledge in their parameters, making the validity of specific facts difficult to check. [44].
- *Unreliable Generation Evaluation* LLMs' evaluation of generated text can be inconsistent and unreliable. Getting consistent results between human judgments and automatic evaluation tools is hard. Even humans do not always agree on the quality of outputs. In addition, the models themselves can be biased based on their training data [93].
- *Under-performing Specialized Generation* LLMs have difficulty performing adequately in specialized jobs due to their lack of domain-specific training. It is challenging to incorporate specific information without diminishing their overall ability [93].
- *Reasoning Inconsistency* An inconsistent result between the reasoning process and the derived solution can arise from LLMs producing the correct answer after taking an improper or incorrect answer after a legitimate reasoning path. Using an ensemble of different reasoning paths, improving the reasoning process, and fine-tuning LLMs with process-level feedback can all help to mitigate the problem [93].
- *Numerical Computation* Numerical calculation presents challenges for LLMs, particularly for infrequently encountered symbols in pre-training. Tokenizing digits into separate tokens and employing mathematical tools is a practical design for improving LLM arithmetic [93].
- *Black-Box Nature* LLMs' decision-making processes are not transparent, making it difficult to comprehend how they arrive at specific predictions or outcomes [44].
- *Indecisiveness* LLMs use probabilistic reasoning, which might result in indecisive or ambiguous results [44].
- *Domain-Specific Knowledge* LLMs trained on generic corpora may not be able to generalize domain-specific or novel knowledge efficiently. [44].

Table 4 Comparison of Key Benchmarks for Evaluating LLMs Integrated with KGs

Benchmark	Task Type	Key Evaluation Metrics	Focus Area
GLUE	Natural language understanding	Accuracy, F1-score	Sentence similarity, entailment, sentiment
SuperGLUE	Advanced natural language understanding	Accuracy, F1-score	Causal reasoning, coreference resolution
SQuAD	Question answering	F1-score, Exact Match	Text comprehension and retrieval
CommonsenseQA	Commonsense reasoning	Accuracy	Commonsense knowledge in QA
WikiKG90M	Knowledge graph completion	Link prediction, entity classification	Graph understanding and completion
Open Graph Benchmark (OGB)	Machine learning on graphs	Node/link/graph classification	Realistic, large-scale graph-based tasks
ATOMIC	Commonsense reasoning	Accuracy	Everyday commonsense knowledge
XNLI	Cross-lingual language understanding	Accuracy	Language inference across languages
HellaSwag	Commonsense reasoning in language	Accuracy	Sentence completion
CLUE	Chinese language understanding	Accuracy, F1-score	Sentence classification, comprehension
ReClor	Logical reasoning in reading comprehension	Accuracy, Logical Consistency	Understanding logical relationships
LAMA	Factual knowledge in language models	Recall	Recall of factual information
T-REx	Text generation from structured data	Accuracy, Fidelity	Text generation from knowledge bases
NELL-995	Knowledge extraction and completion	Accuracy, Completeness	Inference from knowledge base entries
WebQuestionsSP	Question answering over knowledge graphs	Accuracy, Precision	Querying structured data in KGs
ComplexWebQuestions	Complex question answering over KGs	Accuracy, Reasoning Depth	Multi-hop reasoning, compositional queries
MetaQA	Multi-hop question answering over KGs	Accuracy, Reasoning Steps	Multi-step reasoning over structured data
GrailQA	Generalization in KG question answering	Generalization, Accuracy	Unseen entities and relations in KGs
SimpleQuestions	Simple question answering over KGs	Accuracy, Speed	Single-hop queries using KGs
FreebaseQA	Question answering using the Freebase KG	Accuracy, Depth of Knowledge	Large-scale structured data handling

8.2 Challenges related to knowledge graphs

- *Heterogeneous Data* Dealing with different data sources, like web pages, tables, and documents, [46].
- *Evolving Data* Keeping the knowledge graph up-to-date with the latest information [46].
- *Noisy Data* Removing incorrect or irrelevant information to keep the knowledge graph accurate [46].
- *Low-resource Data* Building a complete knowledge graph even when there's not much data available [46].
- *Multi-modal Knowledge Graphs* Combining different data types, such as text and images, into one knowledge graph [46].
- *Cross-lingual Knowledge Graphs* Including data from multiple languages and ensuring they align correctly [46].
- *End-to-end Unified Frameworks* Creating systems that can handle all steps of building and improving a knowledge graph in one go[46].

- *Complex Data* Understanding and managing complicated relationships and contexts in the data [46].
- *Conditional Knowledge* Handling information that changes based on different situations or times [46].
- *Interpretability* Making sure the models are easy to understand and explain their decisions [46].
- *Autonomous Data* Incorporating data created by users, such comments, reviews, and social network posts [46]. Handling diverse and context-specific user-generated data is challenging due to its unstructured nature, slang, acronyms, and contextual references. Advanced techniques are needed to evaluate and integrate this dynamic, updated, and context-specific content into the knowledge network while protecting privacy and addressing individual user biases.

8.3 Challenges related to LMMs and KGs integration

Several challenges exist when integrating knowledge graphs (KGs) with large language models (LLMs), which include but are not limited to:

1. *Computational Resources* Pre-training and fine-tuning LLMs using KGs is computationally demanding, often requiring extensive resources such as high-performance GPUs or TPUs and large memory capacities. The integration process involves training the LLM on vast textual corpora and encoding and embedding complex graph structures, further intensifying the computational requirements. This computational overhead may restrict the feasibility of such integration in resource-constrained environments or real-time applications.
2. *Data Privacy Concerns* Incorporating KGs into LLMs introduces unique privacy challenges. KGs often contain sensitive, domain-specific data (e.g., medical records and personal information) that may require strict privacy controls. When these sensitive datasets are integrated with LLMs, there is a risk of exposing private or confidential information through model outputs, notably if the LLM lacks privacy-preserving mechanisms. Therefore, ensuring that the integrated system adheres to data privacy regulations (such as GDPR) and employs privacy-preserving techniques (e.g., differential privacy) is critical.
3. *Data Dependency and Adaptation Challenges* Fine-tuning LLMs with KGs is most effective when high-quality, specialized datasets are available. However, obtaining and curating domain-specific KGs that are comprehensive and up-to-date is often challenging. This issue is exacerbated in rapidly evolving fields, where the LLM must quickly adapt to new concepts and relationships. Without a continuous pipeline for acquiring and incorporating fresh data, the integrated system's performance may degrade over time, leading to outdated or irrelevant knowledge.
4. *Fact-Checking and Validation Complexity* While integrating KGs with LLMs enhances the factual accuracy of generated content, validating outputs against a KG is not straightforward. Fact-checking requires mapping the generated text to the appropriate entities and relationships in the KG, which is computationally expensive and time-consuming. Furthermore, the fact-checking process might still overlook specific nuances or context-specific inaccuracies, making it challenging to guarantee fully reliable outputs.
5. *Maintaining Accurate and Up-to-Date KGs* The dynamic nature of knowledge, where facts and relationships frequently change, necessitates constant updates to KGs. Maintaining an accurate and up-to-date KG is a complex task that involves automatically extracting, validating, and integrating new information while resolving inconsistencies and redundancies. For LLMs that depend on KGs for accurate reasoning and contextual responses, any lag in updating the KG can negatively impact the relevance and accuracy of generated outputs.

6. *Computational Overhead and Scalability* The scalability of LLMs when integrated with large-scale KGs is a major concern. As KGs grow in size, the computational burden of incorporating all relevant entities and relationships into LLMs becomes significantly higher. Efficient data management strategies, scalable encoding techniques, and model pruning must be implemented to mitigate latency and resource usage. Balancing the trade-off between performance and computational cost is crucial for deploying LLM-KG systems in real-world applications.
7. *Complexity of Graph Reasoning and Inference* While KGs provide structured knowledge, leveraging this structure for reasoning and inference within LLMs is challenging. Unlike textual data, KGs contain interconnected nodes and edges representing complex relationships. Integrating this graph structure into LLMs requires advanced encoding algorithms that capture local and global graph properties, ensuring the model can perform deep reasoning over these relationships.

By addressing these challenges in LLM-KG integration, future research can develop more efficient, reliable, and secure systems that harness the complementary strengths of KGs and LLMs for advanced AI applications.

9 Conclusion and Future Directions

9.1 Summary of key findings and insights

In this paper, the strengths of LLMs and KGs are complementary: the former excels at natural language understanding and generation, and the latter provides structured, factual knowledge that enhances the accuracy and interpretability of AI output. This survey categorized such integration approaches into three principal paradigms: KG-augmented LLMs, LLM-augmented KGs, and synergized frameworks that mutually enhance both technologies. The review demonstrates that complex algorithms for encoding graph data enable the modeling of intricate relationships within knowledge graphs, offering significant benefits for natural language tasks. The key findings of this survey emphasize the importance of adaptive integration techniques tailored to domain-specific requirements. Scalability and performance optimization are crucial for effectively handling large volumes of data and ensuring responsiveness in real-time applications. Moreover, we identified that the effectiveness of these integration approaches is best evaluated using a combination of quantitative metrics (e.g., precision, recall, F1-score) and qualitative assessments (e.g., interpretability, factual consistency, and enrichment capability). These metrics enable a holistic evaluation of how well KGs and LLMs work together to improve generated outputs' accuracy, coherence, and relevance. Another important finding relates to the technical barriers that must be overcome to harness the full potential of KGs for enhancing LLMs' reasoning abilities. Challenges such as computational resource constraints, data dependency, fact-checking, and the quality of knowledge graphs play significant roles in the integration's efficacy. By addressing these barriers, integrating KGs with LLMs can lead to more robust, contextually aware AI systems.

9.2 Future research directions

Further research must refine the methods to seamlessly integrate LLMs with graph databases and other complex data structures. As the survey revealed, sophisticated techniques for data exchange between graph databases and LLMs are paramount. It is crucial to improve encoding algorithms to capture the fine-grained details of relationships in graph data, enabling LLMs to provide more accurate and contextually relevant information. When LLMs are aligned with graph databases specific to a domain, adaptation algorithms will allow better integration and contextualization. Additionally, research should optimize the integration for efficiency and effectiveness by developing scalable, real-time learning models. Such models should dynamically learn from updated KGs, allowing the LLMs to adjust to new data and context swiftly. The scalability and performance challenges in managing the increasing data volumes associated with LLM applications necessitate solutions like model pruning and efficient architectures. These solutions can reduce latency while maintaining or even enhancing model performance. Furthermore, enhancing evaluation techniques for integrated systems is a key direction, aiming to comprehensively measure not only traditional performance metrics but also more complex aspects like knowledge representation and reasoning capabilities. Bias mitigation is another critical research avenue. Domain-specific KGs have the potential to identify and reduce biases in LLM outputs. By developing techniques that ensure factual consistency and fairness, future integrated systems can generate more reliable, transparent, and unbiased outputs. Finally, there are opportunities for creating interdisciplinary approaches that combine the strengths

of AI, NLP, and database technologies. Such approaches would advance techniques for real-time learning, efficient data management, and seamless knowledge transfer between KGs and LLMs, further enhancing the capabilities and applications of these integrated systems.

Data availability There is no data used in this paper.

References

1. Wang H, Xu Z, Fujita H, Liu S. Towards felicitous decision making: An overview on challenges and trends of big data. Inform Sci. 2016;367:747–65.
2. Hu H, Wen Y, Chua T-S, Li X. Toward scalable systems for big data analytics: a technology tutorial. IEEE Access. 2014;2:652–87.
3. Yang J, Yao W, Zhang W. Keyword search on large graphs: A survey. Data Sci Eng. 2021;6(2):142–62.
4. Yuan Y, Lian X, Chen L, Yu JX, Wang G, Sun Y. Keyword search over distributed graphs with compressed signature. IEEE Trans Knowle Data Eng. 2017;29(6):1212–25.
5. Brown TB. Language models are few-shot learners. arXiv preprint arXiv:2005.14165 2020.
6. Lee J, Toutanova K. Pre-training of deep bidirectional transformers for language understanding. arXiv preprint 2018. arXiv:1810.04805.
7. Jiang J, Huang X, Choi B, Xu J, Bhowmick SS, Xu L. ppkws: an efficient framework for keyword search on public-private networks. In: 2020 IEEE 36th International Conference on Data Engineering (ICDE), 2020.
8. Hogan A, Blomqvist E, Cochez M, d'Amato C, Melo GD, Gutierrez C, Kirrane S, Gayo JEL, Navigli R, Neumaier S, et al. Knowledge graphs. ACM Comput Surveys. 2021;54(4):1–37.
9. Bollacker K, Evans C, Paritosh P, Sturge T, Taylor J. Freebase: A collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, 2008.
10. Lully V, Laublet P, Stankovic M, Radulovic F. Enhancing explanations in recommender systems with knowledge graphs. Proc Comput Sci. 2018;137:211–22.
11. Church KW. Word2vec. Nat Lang Eng. 2017;23(1):155–62.
12. Pennington J, Socher R, Manning CD. Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014. .
13. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. Adv Neural Inform Proc Syst 2017;30.
14. Roumeliotis KI, Tselikas ND. Chatgpt and open-ai models: a preliminary review. Future Internet. 2023;15(6):192.
15. Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, Zhou Y, Li W, Liu PJ. Exploring the limits of transfer learning with a unified text-to-text transformer. J Machine Learn Res. 2020;21(140):1–67.
16. Chowdhery A, Narang S, Devlin J, Bosma M, Mishra G, Roberts A, Barham P, Chung HW, Sutton C, Gehrmann S, et al. Palm: scaling language modeling with pathways. J Machine Learn Res. 2023;24(240):1–113.
17. Team G, Anil R, Borgeaud S, Wu Y, Alayrac J-B, Yu J, Soricut R, Schalkwyk J, Dai AM, Hauth A et al. Gemini: a family of highly capable multimodal models. arXiv preprint. 2023 arXiv:2312.11805.
18. Liu Z, Lin W, Shi Y, Zhao J. A robustly optimized bert pre-training approach with post-training China National Conference on Chinese Computational Linguistics, . Springer. 2021.
19. Zhang S, Roller S, Goyal N, Artetxe M, Chen M, Chen S, Dewan C, Diab M, Li X, Lin XV et al. Opt: Open pre-trained transformer language models. arXiv preprint. 2022. arXiv:2205.01068.
20. Touvron H, Lavril T, Izacard G, Martinet X, Lachaux M-A, Lacroix T, Rozière B, Goyal N, Hambro E, Azhar F et al. Llama: Open and efficient foundation language models. arXiv preprint. 2023. arXiv:2302.13971.
21. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I, et al. Language models are unsupervised multitask learners. OpenAI blog. 2019;1(8):9.
22. Gallifant J, Fiske A, Levites Strekalova YA, Osorio-Valencia JS, Parke R, Mwavu R, Martinez N, Gichoya JW, Ghassemi M, Demner-Fushman D, et al. Peer review of gpt-4 technical report and systems card. PLOS Digital Health. 2024;3(1):0000417.
23. Lewis M. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint. 2019. arXiv:1910.13461
24. Chen M, Tworek J, Jun H, Yuan Q, Pinto HPDO, Kaplan J, Edwards H, Burda Y, Joseph N, Brockman G et al. Evaluating large language models trained on code. arXiv preprint. 2021. arXiv:2107.03374.
25. Minaee S, Mikolov T, Nikzad N, Chenaghlu M, Socher R, Amatriain X, Gao J. Large language models: A survey. arXiv preprint. 2024. arXiv:2402.06196.

26. Bender EM, Gebru T, McMillan-Major A, Shmitchell S. On the dangers of stochastic parrots: Can language models be too big? In: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, 2021.
27. Weidinger L, Mellor J, Rauh M, Griffin C, Uesato J, Stiennon N, Gabriel I. Ethical and social risks of harm from language models. arXiv preprint. 2021. arXiv:2110.01134.
28. Singhal A. Introducing the Knowledge Graph: things, not strings. 2012; https://blog.google/products/search/introducing-knowledge-graph-things-not/. Accessed: 11 7 2024
29. Auer S, Bizer C, Kobilarov G, Lehmann J, Cyganiak R, Ives Z. Dbpedia: a nucleus for a web of open data. The Semantic Web. 2007. https://doi.org/10.1007/978-3-540-76298-0_52.
30. Chang E, Mostafa J. The use of snomed ct, 2013–2020: a literature review. J Am Med Inform Assoc. 2021;28(9):2017–26.
31. Bennett M. The financial industry business ontology: best practice for big data. J Bank Reg. 2013;14(3):255–68.
32. Zhao X, Wang Y, Qin J, Gao L. Supplychainkg: An event-based knowledge graph generator for supply chain management. In: Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM), 2020.
33. Paulheim H. Knowledge graph refinement: a survey of approaches and evaluation methods. Semantic web. 2017;8(3):489–508.
34. Vrandečić D, Krötzsch M. Wikidata: a free collaborative knowledge base. Commun ACM. 2014;57(10):78–85. https://doi.org/10.1145/2629489.
35. Bizer C, Heath T, Berners-Lee T. Linked data-the story so far. In: Linking the World's Information: Essays on Tim Berners-Lee's Invention of the World Wide Web, 2023; p. 115–143.
36. Berners-Lee T, Hendler J, Lassila O. The semantic web. Sci Am. 2001;284(5):34–43. https://doi.org/10.1038/scientificamerican0501-34.
37. Fensel D, Simsek U, Angele K, Huaman E, Kärle E, Panasiuk O, Toma I, Umbrich J, Wahler A. Knowle Graphs. Berlin: Springer; 2020.
38. Guo Q, Zhuang F, Qin C, Zhu H, Xie X, Xiong H, He Q. A survey on knowledge graph-based recommender systems. IEEE Trans Knowle Data Eng. 2020;34(8):3549–68.
39. Peng C, Xia F, Naseriparsa M, Osborne F. Knowledge graphs: opportunities and challenges. Artif Intell Rev. 2023;56(11):13071–102.
40. Düggelin W, Laurenzi E. A knowledge graph-based decision support system for resilient supply chain networks. In: International Conference on Research Challenges in Information Science. Springer. 2024.
41. Wang H, Zheng J, Carvajal-Roca IE, Chen L, Bai M. Financial fraud detection based on deep learning: Towards large-scale pre-training transformer models. In: China Conference on Knowledge Graph and Semantic Computing, 2023. Springer.
42. Przysucha B, Kaleta P, Dmowski A, Piwkowski J, Czarnecki P, Cieplak T. Product knowledge graphs: creating a knowledge system for customer support. 2024.
43. Ji S, Pan S, Cambria E, Marttinen P, Yu PS. A survey on knowledge graphs: representation, acquisition, and applications. IEEE Trans Neural Net Learn Syst. 2021;33(2):494–514.
44. Pan S, Luo L, Wang Y, Chen C, Wang J, Wu X. Unifying large language models and knowledge graphs: a roadmap. IEEE Transactions on Knowledge and Data Engineering. 2024.
45. Yang J, Hu X, Xiao G, Shen Y. A survey of knowledge enhanced pre-trained models. arXiv preprint. 2021. arXiv:2110.00269
46. Zhong L, Wu J, Li Q, Peng H, Wu X. A comprehensive survey on automatic knowledge graph construction. ACM Comput Surv. 2023;56(4):1–62.
47. Wang X, Gao T, Zhu Z, Liu Z, Li J, Kepler JT. A unified model for knowledge embedding and pre-trained language representation. Trans Assoc Comput Linguistics. 2021;9:176–94. https://doi.org/10.1162/tacl_a_00360.
48. Moon C, Jones P, Samatova NF. Learning entity type embeddings for knowledge graph completion. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 2017.
49. Li D, Xu F. Synergizing knowledge graphs with large language models: a comprehensive review and future prospects. arXiv preprint. 2024. arXiv:2407.18470.
50. Hamilton WL, Ying R, Leskovec J. Representation learning on graphs: methods and applications. arXiv preprint. 2017. arXiv:1709.05584.
51. Xia F, Sun K, Yu S, Aziz A, Wan L, Pan S, Liu H. Graph learning: a survey. IEEE Trans Artif Intell. 2021;2(2):109–27.
52. Zhang Y, Dai H, Kozareva Z, Smola A, Song L. Variational reasoning for question answering with knowledge graph. Proc AAAI Conf Artif Intell. 2018;32:1.
53. Lully V, Laublet P, Stankovic M, Radulovic F. Enhancing explanations in recommender systems with knowledge graphs. Procedia Comput Sci. 2018;137:211–22.
54. Hofer M, Obraczka D, Saeedi A, Köpcke H, Rahm E. Construction of knowledge graphs: current state and challenges. Information. 2024;15(8):509.
55. Chen Z, Mao H, Li H, Jin W, Wen H, Wei X, Wang S, Yin D, Fan W, Liu H, et al. Exploring the potential of large language models (llms) in learning on graphs. ACM SIGKDD Explorations Newsletter. 2024;25(2):42–61.
56. Liang Y, Tan K, Xie T, Tao W, Wang S, Lan Y, Qian W. Aligning large language models to a domain-specific graph database. arXiv preprint. 2024. arXiv:2402.16567.
57. Fatemi B, Halcrow J, Perozzi B. Talk like a graph: encoding graphs for large language models. arXiv preprint. 2023. arXiv:2310.04560.
58. Zhou X, Sun Z, Li G. Db-gpt: large language model meets database. Data Sci Eng. 2024;1:10.
59. Colombo-Mendoza LO, Valencia-García R, Rodríguez-González A, Colomo-Palacios R, Alor-Hernández G. Towards a knowledge-based probabilistic and context-aware social recommender system. J Inform Sci. 2018;44(4):464–90.
60. Lewis P, Oguz B, Rinott R, Riedel S, Stoyanov V. Retrieval-augmented generation for knowledge-intensive nlp tasks. Adv Neural Inform Proc Syst. 2020;33:9459–74.
61. Liu PJ, Saleh M, Pot E, Goodrich B, Sepassi R, Kaiser L, Shazeer N. Generating wikipedia by summarizing long sequences. arXiv preprint. 2018. arXiv:1801.10198.
62. Pokornỳ J. Integration of relational and graph databases functionally. Found Comput Dec Sci. 2019;44(4):427–41.
63. Karpukhin V, Oguz B, Min S, Lewis P, Wu L, Edunov S, Yih W-t. Dense passage retrieval for open-domain question answering. arXiv preprint. 2020. arXiv:2004.04906.
64. Izacard G, Grave E. Leveraging passage retrieval with generative models for open domain question answering. arXiv preprint. 2020. arXiv:2007.01282.

65. Zhao P, Zhang H, Yu Q, Wang Z, Geng Y, Fu F, Cui B. Retrieval-augmented generation for ai-generated content: A survey. arXiv preprint. 2023. arXiv:2402.19473.
66. Zhang X, Ju T, Liang H, Fu Y, Zhang Q. Llms instruct llms: an extraction and editing method. arXiv preprint. 2024. arXiv:2403.15736.
67. Ji S, Pan S, Cambria E, Marttinen P, Philip SY. A survey on knowledge graphs: representation, acquisition, and applications. IEEE Trans Neural Net Learn Syst. 2021;33(2):494–514.
68. Cámara J, Troya J, Burgueño L, Vallecillo A. On the assessment of generative ai in modeling tasks: an experience report with chatgpt and uml. Software Syst Mod. 2023;22(3):781–93.
69. Bommasani R, Hudson DA, Adeli E, Altman R, Arora S, Arx S, Liang P. On the opportunities and risks of foundation models. arXiv preprint. 2021. arXiv:2108.07258.
70. Agrawal G, Kumarage T, Alghami Z, Liu H. Can knowledge graphs reduce hallucinations in llms?: A survey. arXiv preprint. 2022. arXiv:2311.07914.
71. Rosset C, Xiong C, Phan M, Song X, Bennett P, Tiwary S. Knowledge-aware language model pretraining. arXiv preprint. 2020. arXiv:2007.00655.
72. Eppalapally S, Dangi D, Bhat C, Gupta A, Zhang R, Agarwal S, Bagga K, Yoon S, Lipka N, Rossi RA et al. Kapqa: Knowledge-augmented product question-answering. arXiv preprint. 2024. arXiv:2407.16073.
73. Wang A. Glue: A multi-task benchmark and analysis platform for natural language understanding. arXiv preprint. 2018. arXiv:1804.07461.
74. Wang A, Pruksachatkun Y, Nangia N, Singh A, Michael J, Hill F, Levy O, Bowman SR. Superglue: a stickier benchmark for general-purpose language understanding systems. Proceedings of NeurIPS. 2019.
75. Rajpurkar P, Zhang J, Lopyrev K, Liang P. Squad: 100,000+ questions for machine comprehension of text. arXiv preprint. 2016. arXiv:1606.05250.
76. Talmor A, Herzig J, Lourie N, Berant J. Commonsenseqa: a question answering challenge targeting commonsense knowledge. arXiv preprint. 2018. arXiv:1811.00937.
77. Kochsiek A, Gemulla R. A benchmark for semi-inductive link prediction in knowledge graphs. arXiv preprint. 2023. arXiv:2310.11917.
78. Hu W, Fey M, Zitnik M, Dong Y, Ren H, Liu B, Catasta M, Leskovec J. Open graph benchmark: datasets for machine learning on graphs. Proceedings of NeurIPS 2020.
79. Sap M, Bras RL, Allaway E, Bhagavatula C, Lourie N, Rashkin H, Roof B, Smith NA, Choi Y. Atomic: an atlas of machine commonsense for if-then reasoning. Proceedings of AAAI 2019.
80. Conneau A, Lample G, Rinott R, Williams A, Bowman SR, Schwenk H, Stoyanov V. Xnli: Evaluating cross-lingual sentence representations. arXiv preprint. 2018. arXiv:1809.05053.
81. Zellers R, Holtzman A, Bisk Y, Farhadi A, Choi Y. Hellaswag: Can a machine really finish your sentence? arXiv preprint. 2019. arXiv:1905.07830.
82. Xu L, Hu H, Zhang X, Li L, Cao C, Li Y, Xu Y, Sun K, Yu D, Yu C et al. Clue: A chinese language understanding evaluation benchmark. arXiv preprint. 2020. arXiv:2004.05986.
83. Yu W, Jiang Z, Dong Y, Feng J. Reclor: A reading comprehension dataset requiring logical reasoning. arXiv preprint. 2020. arXiv:2002.04326.
84. Petroni F, Rocktäschel T, Lewis P, Bakhtin A, Wu Y, Miller AH, Riedel S. Language models as knowledge bases? arXiv preprint. 2019. arXiv:1909.01066.
85. Elsahar H, Vougiouklis P, Remaci A, Gravier C, Hare J, Laforest F, Simperl E. T-rex: A large scale alignment of natural language with knowledge base triples. Proceedings of LREC; 2018.
86. Carlson A, Betteridge J, Kisiel B, Settles B, Hruschka ER, Mitchell TM. Toward an architecture for never-ending language learning. Proceedings of AAAI, 2010.
87. Berant J, Liang P. Semantic parsing via paraphrasing, 2014;1415–1425.
88. Talmor A, Berant J. The web as a knowledge-base for answering complex questions. arXiv preprint. 2018. arXiv:1803.06643.
89. Zhang Y, Dai H, Kozareva Z, Smola A, Song L. Variational reasoning for question answering with knowledge graph. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2018.
90. Jiang L, Usbeck R. Knowledge graph question answering datasets and their generalizability: are they enough for future research? In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022.
91. Bordes A, Usunier N, Chopra S, Weston J. Large-scale simple question answering with memory networks. arXiv preprint. 2015. arXiv:1506.02075.
92. Fader A, Zettlemoyer L, Etzioni O. Open question answering over curated and extracted knowledge bases. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014.
93. Zhao WX, Zhou K, Li J, Tang T, Wang X, Hou Y, Min Y, Zhang B, Zhang J, Dong Z et al. A survey of large language models. arXiv preprint. 2023. arXiv:2303.18223.