SBC Computing Reviews, 2026, 05:1
ISSN: 2966-3938 • doi: 10.5753/reviews.2026.6738

Review

Constructing Knowledge Graphs from Text Using Large Language Models: Scoping Review

Giovanna Borges Bottino [Universidade de São Paulo | *giovanna.bottino@usp.br*]
José de Jesus Pérez Alcazár [Universidade de São Paulo | *jperez@usp.br*]

School of Arts, Sciences and Humanities, University of São Paulo, Arlindo Béttio street, 1000, São Paulo, SP, 03828-000, Brazil.

Abstract. Constructing Knowledge Graphs from unstructured textual sources poses significant challenges due to the inherent ambiguity of natural language and the high cost and limited scalability of manual knowledge modeling. Recently, Large Language Models (LLMs) have emerged as a promising alternative for automating knowledge extraction and structuring, resulting in a rapidly expanding body of research. This article aims to provide a scoping review of methods using LLMs to construct Knowledge Graphs from text, map existing approaches, identify methodological patterns, and analyze their strengths and limitations, thereby synthesizing the state of the art. The review employs a systematic protocol consistent with PRISMA guidelines and examines 126 primary studies. The literature is categorized into four methodological groups: Ontology-Based, Prompt-Based, RAG-Based, and Hybrid Pipelines. Each category is analyzed with respect to the role of LLMs within the construction pipeline, the degree of semantic formalization, and the architectural strategies implemented. Comparative analysis is performed using five evaluation criteria: exactness, scalability, adaptability, reproducibility, and ease of implementation. The findings demonstrate that no single approach comprehensively addresses all challenges inherent to LLM-based Knowledge Graph construction. Ontology-Based methods provide robust semantic guarantees but demand considerable manual effort. Prompt-Based approaches facilitate rapid deployment yet exhibit variability and limited reproducibility. RAG-Based methods enhance grounding through external evidence but introduce reliance on retrieval mechanisms. Hybrid Pipelines yield higher extraction accuracy, though at the expense of increased architectural complexity. The review identifies substantial heterogeneity in evaluation practices and a lack of standardized quantitative metrics. These results underscore the necessity for consistent evaluation frameworks to enhance comparability, reliability, and the advancement of research in LLM-assisted Knowledge Graph construction.

Keywords: Knowledge Graphs, Large Language Models, Scoping Review

Received: 22 October 2025 • **Accepted:** 15 June 2026 • **Published:** 25 August 2026

1 Introduction

Organizations produce substantial documentation during projects, such as financial reports, employee performance records, and design and engineering documents. Approximately 90% of this information is unstructured. Despite its prevalence, organizations frequently lack comprehensive insight into the content of these documents, even though about 58% of unstructured data is reused after its initial application [Muscolino *et al*., 2023; Negro *et al*., 2022].

Unstructured data represents a significant information asset, and organizations capable of processing, managing, and analyzing it may achieve a competitive advantage. Utilizing this data enhances productivity and helps manage growing data complexity. Moreover, expressing a rich semantic model in a machine-readable format enables the derivation of advanced insights. It supports complex reasoning, such as causal inference [Negro *et al*., 2022; Kejriwal *et al*., 2021].

Transforming unstructured data into structured knowledge is a complex, multi-stage process. The inherent challenges associated with enabling machines to understand natural language have contributed to the adoption of Knowledge Graphs (KGs). A Knowledge Graph provides a practical, machine-readable representation of information about the world, consisting of interconnected facts that describe real-world entities and their relationships. This representation allows knowledge to be interpreted by both humans and computational systems [Negro *et al*., 2022; Kejriwal *et al*., 2021].

Knowledge Graphs have demonstrated their usefulness in enriching business processes by enabling contextualized content delivery and personalized responses. However, constructing knowledge graphs remains challenging, and developing systematic solutions that automatically generate them from unstructured data would constitute a significant advancement over predominantly manual approaches [Zhong *et al*., 2023].

The automatic construction of a knowledge graph requires advanced techniques to represent entities and relationships within a network structure. This process typically involves entity recognition, coreference resolution, and relation extraction. Entity recognition focuses on identifying mentions of entities in the data, coreference resolution determines which mentions refer to the same entity, and relation extraction establishes semantic connections between entities [Negro *et al*., 2022; Zhong *et al*., 2023].

In this context, Large Language Models (LLMs) offer a faster, more efficient alternative to manual knowledge graph creation. These models can handle spelling variations, inconsistencies in entity naming, and incomplete information, while also capturing the complex linguistic contexts present in natural language text [Negro *et al*., 2022; Alammar and

Grootendorst, 2024].

As Large Language Models increasingly automate knowledge extraction and structuring, understanding the methodologies for constructing Knowledge Graphs from textual data is essential. A scoping review is warranted to systematically map current techniques, evaluate their effectiveness, and identify gaps in the literature. This analysis informs the selection of suitable methodologies, supports the refinement of existing approaches, and advances research and practice in the field.

The structure of this article is as follows. The first section, *Basic Concepts*, establishes the conceptual foundation for the research. The second section, *Related Surveys*, reviews prior surveys in the field and delineates the primary contributions of this study. The third section, *Methodology*, details the research protocol, including the review process and reporting strategy. The article then analyzes identified knowledge graph construction methods and concludes with a discussion of the principal findings.

2 Basic Concepts

This section outlines the conceptual foundations underpinning the study. It first examines the principles and challenges of knowledge graph construction, with emphasis on its incremental and use-oriented characteristics. It then introduces Large Language Models, discussing their core mechanisms, interaction paradigms, and related techniques, such as prompt engineering, retrieval-augmented generation, and fine-tuning strategies. Collectively, these discussions provide the conceptual basis for analyzing the application of LLMs in knowledge graph construction.

2.1 Knowledge Graph Construction

Knowledge graph construction is a complex task, particularly because there is rarely a pre-existing graph that fully meets the needs of a specific domain. As a result, it is commonly understood as an incremental process in which heterogeneous data are integrated into a unified, coherent, and semantically grounded representation. This incremental nature implies that conceptual and modeling decisions are continuously revised as new data sources or new analytical demands emerge [Kejriwal *et al*., 2021; Negro *et al*., 2022].

In this context, knowledge graph construction has been widely investigated as an alternative to manual approaches, which are traditionally characterized by high cost, low scalability, and maintenance difficulties. More specifically, this process is also incremental, whereby data from heterogeneous sources are progressively transformed into a semantic structure composed of entities, relations, and explicit descriptions [Zhong *et al*., 2023].

Along these lines, Negro *et al*. [2022] emphasizes that automatic knowledge graph construction should be understood as a layered and use-case-driven process. Rather than focusing initially on defining complex semantic structures, the emphasis lies on the explicit organization of entities, relations, and properties extracted from data. Notably, the authors highlight that outputs from language models or automatic extraction techniques should not be treated as the final graph. Instead, they should be treated as intermediate representations, subject to normalization, consolidation, and reorganization before their definitive incorporation into the knowledge graph.

From this perspective, it becomes essential to separate the extraction stages from the graph organization stages. While extraction focuses on automatically identifying relevant elements in texts, graph organization aims to ensure structural consistency, reduce redundancy, and maintain information traceability to the original data sources. Consequently, this separation enhances transparency in the construction process and enables iterative adjustments as the graph evolves [Negro *et al*., 2022].

Therefore, automatic knowledge graph construction should not be understood as a linear or one-off procedure, but rather as a continuous refinement process. In this process, the adopted level of structure is determined by usage needs and analytical demands. Effective knowledge graphs are thus those whose degree of semantic organization is compatible with their objectives, balancing structural simplicity, interpretability, and the potential for long-term evolution [Zhong *et al*., 2023; Negro *et al*., 2022].

2.2 Large Language Models

In parallel with advances in knowledge graph construction, Large Language Models have emerged as a central technological component in contemporary natural language processing. These models consist of computational systems trained to predict the next textual unit in a sequence, where this unit is defined at the token level. Learning takes place using extensive collections of previously tokenized text, in which the model adjusts its parameters to minimize the error between predicted and observed tokens. Through this self-supervised learning process, models can capture statistical regularities of language, including syntactic patterns, semantic associations, and discourse structures [Jurafsky and Martin, 2025].

Text generation in LLMs is conditional, as each generated sequence depends on the initial context provided to the system. Within this process, three concepts play a central role: prompt, temperature, and Top-K [Jurafsky and Martin, 2025].

- The prompt corresponds to the initial text presented to the model and defines the context in which subsequent tokens will be predicted.
- temperature controls the degree of randomness in selecting the next token; increasing it broadens the probability distribution.
- The Top-K method restricts the selection to the most probable tokens, limiting variability and increasing determinism as the value of K decreases.

Taken together, these mechanisms regulate how the model's knowledge is mobilized to produce coherent and contextually appropriate texts. In the following section, the concepts of prompt engineering, Retrieval-Augmented Generation (RAG), and fine-tuning are defined.

2.2.1 Prompt Engineering

Given that interaction with Large Language Models primarily occurs through natural language, the formulation of adequate instructions is a critical factor. For this reason, prompt

engineering has emerged as a systematic practice for designing textual prompts that guide the model's behavior during response generation [Berryman and Ziegler, 2024; Phoenix and Taylor, 2024].

In general, the prompt acts as the primary conditioning mechanism for generation. It may take the form of direct questions, explicit instructions, or more elaborate structures combining context, constraints, and other elements. In this sense, instructions formulated clearly and explicitly tend to produce responses more closely aligned with user expectations, particularly when they precisely delimit both the task and the desired output format [Berryman and Ziegler, 2024].

Beyond direct instructions, prompt engineering may also incorporate examples to induce specific response patterns. This approach, known as in-context learning, allows the model to identify regularities from the examples provided within the prompt itself and to adjust its generation accordingly [Berryman and Ziegler, 2024].

Furthermore, the position of information within the prompt plays a relevant role. Information placed at the beginning or end of the prompt tends to exert greater influence on the generated response. In contrast, content in the middle may become less relevant, a phenomenon referred to as middle-content loss [Berryman and Ziegler, 2024].

Additionally, prompts structured into well-defined components, such as context, contexttion, examples, and response format, tend to yield greater consistency and predictability, particularly in more complex applications [Mao *et al*., 2025]. Thus, prompt engineering encompasses not only lexical choices but also structural decisions regarding information ordering and explicit articulation of interaction objectives.

In this work, four main types of prompt construction were considered: zero-shot, few-shot, sequential prompting chains, and prompt templates.

In zero-shot prompting, the model receives only the primary instruction, without additional examples to guide the expected response pattern. As a consequence, this approach relies heavily on the clarity and precision of textual formulation and on the knowledge previously embedded in the model during training [Schulhoff *et al*., 2024].

By contrast, few-shot prompting incorporates explicit examples within the prompt, guiding the model toward the expected semantic or structural pattern. This technique is beneficial when the central term of the prompt is ambiguous [Reynolds and McDonell, 2021].

Sequential prompting chains further extend this interaction by organizing it into successive stages, in which each response feeds into the subsequent instruction. This strategy enables the decomposition of complex tasks into smaller and more controllable operations [Schulhoff *et al*., 2024].

Finally, prompt templates prioritize structural standardization by using fixed, reusable instruction formats. Unlike sequential prompting chains, which emphasize temporal order, prompt templates focus on predefined components such as the role assigned to the model, the task to be performed, semantic constraints, and the expected response format [Phoenix and Taylor, 2024; Mao *et al*., 2025].

2.2.2 Retrieval-Augmented Generation

Large Language Models can produce responses through conditional generation, leveraging knowledge previously embedded in their parameters during training on large textual corpora. However, this capability presents essential limitations, such as the production of factually incorrect responses, the absence of reliable calibration mechanisms, and the inability to access proprietary or updated information after the training period [Jurafsky and Martin, 2025].

In light of these constraints, integrating Large Language Models with information retrieval techniques emerges as a relevant methodological alternative. Information retrieval is characterized as the process of selecting the most relevant documents from a collection, given a query that expresses an information need [Jurafsky and Martin, 2025].

Retrieval-Augmented generation constitutes an approach that explicitly integrates information retrieval techniques with Large Language Models, aiming to overcome limitations inherent to purely parameter-based generation [Jurafsky and Martin, 2025].

RAG's operation is generally organized into two articulated stages. Initially, given a natural language query, the system triggers a retrieval mechanism that identifies potentially relevant documents or text passages within a predefined collection. This retrieval may be based on sparse representations grounded in term matching or on dense representations in which queries and documents are projected into a shared vector space. In both cases, the objective is to select content that exhibits greater semantic proximity to the expressed information need [Jurafsky and Martin, 2025].

Subsequently, the retrieved texts are incorporated into the input context of the Large Language Model. Response generation then occurs conditionally, not only on the original query but also on the textual evidence provided by the retrieval stage. In this way, the model ceases to operate as a closed system and instead functions as a component integrated with an external document base, using explicit information to support its responses [Jurafsky and Martin, 2025].

2.2.3 fine-tuning

As previously presented, Large Language Models, after the initial pretraining stage, exhibit relevant limitations in their ability to follow human instructions and to produce responses aligned with utility and safety criteria. This limitation occurs because the core objective of pretraining is to predict the next textual unit, which, by itself, does not ensure that the model adequately fulfills the intentions expressed in a request. In this context, a contextual set of training procedures, referred to as post-training, is necessary, within which fine-tuning is situated. fine-tuning is the technique of adapting a trained model to a specific task by retraining it [Jurafsky and Martin, 2025].

Different fine-tuning strategies share the general objective of adapting or aligning model behavior to specific usage requirements. These strategies differ in purpose, data type, and impact on model parameters. In general, three main approaches can be identified: fine-tuning as a continuation of pretraining, aimed at adaptation to new textual domains; instruction-supervised fine-tuning, directed at improving the ability to follow natural language commands; and parameter-

efficient fine-tuning, which seeks to reduce computational costs through partial architectural updates [Jurafsky and Martin, 2025].

Within this set of approaches, instruction-supervised fine-tuning plays a central role in aligning models with human expectations. Instruction fine-tuning consists of continuing the training of a previously trained model using a dataset of natural language instructions paired with correct responses. Unlike pretraining, in which data lack an explicit supervised objective, at this stage each instruction is associated with a reference response, characterizing the process as supervised. Nevertheless, the optimization objective remains the same: predicting the next textual unit using a cross-entropy-based loss function [Jurafsky and Martin, 2025].

3 Related Surveys

This survey is related to the state of the art in constructing Knowledge Graphs from text using Large Language Models. An evaluation of several research survey papers on this theme was conducted. In this section, we present recently proposed survey papers on these topics. The following surveys were identified using the search strategy described in Section 4.1.

The survey by Zong *et al*. [2024] provides a comprehensive overview of challenges in Chinese biomedical text mining, covering the period from 2017 to 2023. It analyzes 39 evaluation tasks in six community challenges, including the China Conference on Knowledge Graph and Semantic Computing and the China Health Information Processing Conference. Furthermore, the study compares these challenges with international initiatives such as BioCreative and discusses how Large-Scale Language Models can improve biomedical knowledge extraction.

Pan *et al*. [2024] provides a roadmap for integrating LLMs and Knowledge Graphs. The authors categorize existing approaches into three principal axes: LLMs using KGs to obtain more reliable information; KGs constructed from text using LLMs; and hybrid systems combining symbolic learning and neural connections.

Li and Xu [2024] presents an overview of recent advancements and challenges in the integration of Knowledge Graphs and Large Language Models. Although the paper discusses how LLMs can contribute to KG construction, especially from text, it does not follow a systematic methodology and lacks a taxonomy of the reviewed methods.

Wang *et al*. [2024] surveys the state of the art in Knowledge Graph construction techniques, including those based on LLMs. The study provides a categorization of construction strategies, such as rule-based, machine learning-based, and LLM-based approaches, and discusses future research directions. However, it does not define explicit research questions nor apply a systematic review protocol.

Surveys by Liang *et al*. [2024] and Zhu *et al*. [2024] were analyzed during an exploratory search. The work of Liang *et al*. [2024] presents a review of LLM-augmented Knowledge Graphs in complex product design, analyzing 42 studies published between 2021 and 2024. While it identifies applications, challenges, and future research directions, it does not cover all methods for constructing Knowledge Graphs from unstructured text.

Similarly, Zhu *et al*. [2024] provides a comprehensive evaluation of LLMs for Knowledge Graph construction and reasoning, offering both quantitative benchmarking and new approaches. Although the paper evaluates LLMs for KG-related tasks, it does not present a structured taxonomy of LLM-driven approaches to KG construction from text.

Table 1 summarizes the surveys analyzed in this section alongside the present work. Each criterion reflects a specific aspect of how survey studies analyze Knowledge Graph construction using Large Language Models. The criterion *Analyzes KG Construction from Text using LLMs* indicates whether a survey explicitly addresses the use of LLMs for constructing Knowledge Graphs from textual data. *Uses Research Questions for Evaluation & comparison* refers to the presence of explicitly stated research questions guiding the comparative analysis. *Follows a Systematic Review Approach (e.g., PRISMA)* captures whether a formal and reproducible review protocol is adopted. *Provides a structured taxonomy of KG construction methods from text using LLM* denotes whether the survey organizes the reviewed methods into a coherent classification scheme. The criterion *Examines the role of LLMs across KG construction stages* reflects whether the survey analyzes how LLMs are applied at different stages of the construction process, such as extraction or structuring. Finally, *Centers on KG construction from unstructured text using LLMs* indicates whether this problem constitutes the primary analytical focus of the survey rather than a secondary or supporting topic. Each criterion is marked by an 'X' when applicable.

This study exists to complement and address key gaps in previous surveys, as highlighted in the comparison table. By centering the analysis on KG construction from unstructured text using LLMs and examining how LLMs are used across the stages of KG construction, this study emphasizes aspects of the construction process while maintaining a systematic approach, research-driven evaluation, and a formal taxonomy.

4 Methodology

This study conducted a scoping review to identify key concepts, available evidence types, and gaps in existing research, thereby mapping the field of study. The review followed the structured methodology outlined by Peters *et al*. [2015], which included defining the objective and guiding research question, establishing inclusion and exclusion criteria, selecting data sources, collecting and organizing the results, and presenting the findings.

The review process was divided into three phases: planning, which involved developing the research protocol; execution, which entailed conducting the review; and reporting, which focused on summarizing and presenting the results.

4.1 Research Protocol

The research protocol for this state-of-the-art review aimed to outline the process for identifying and analyzing the existing body of literature. Below, we describe the research questions, steps, and criteria used to conduct the review.

Table 1. Comparison of Survey Studies on LLMs and Knowledge Graph Construction

Criteria	Zong *et al.* [2024]	Pan *et al.* [2024]	Liang *et al.* [2024]	Zhu *et al.* [2024]	Li and Xu [2024]	Wang *et al.* [2024]	This Study
Analyzes KG Construction from Text using LLMs	-	-	X	X	X	X	X
Uses Research Questions for Evaluation & Comparison	-	X	X	X	-	-	X
Follows a Systematic Review Approach (e.g., PRISMA)	-	-	-	-	-	-	X
Provides a structured taxonomy of KG construction methods from text using LLM	-	-	-	-	-	X	X
Examines the role of LLMs across KG construction stages	-	-	-	-	-	-	X
Centers on KG construction from unstructured text using LLMs	-	-	-	-	-	-	X

4.1.1 Research Questions

The study sought to identify primary research and examine existing approaches in the literature that construct Knowledge Graphs from text sources using Large Language Models. The key research question was framed using the PCC acronym (P = Population; C = Concept; C = context), where the Population is Knowledge Graphs, the Concept is Large Language Models, and the context is raw text as a data source [Peters *et al*., 2015]:

"What techniques are associated with the use of LLMs in constructing knowledge graphs from text sources?"

To guide this review, the following research questions were formulated, each with specific motivations, objectives, and methodological strategies:

Which LLMs have been most commonly used for this purpose?

This question aimed to identify the most frequently used LLMs by analyzing their mentions in primary studies. The objective was to determine which models have been most widely adopted for knowledge graph construction.

In which domains have LLMs been most frequently applied for this purpose?

This research question aimed to identify the domains in which knowledge graph construction with LLMs has been applied. Since knowledge graphs are used across various fields, it was essential to categorize and highlight the most relevant domains reported in the literature.

What are the key advantages of using LLMs for this purpose?

In addition to comparative analysis, this question evaluated the feasibility and benefits of employing LLMs for knowledge graph construction. The findings were synthesized from the reports and insights presented in the primary studies.

What are the key disadvantages of using LLMs for this purpose?

This question focused on identifying potential drawbacks of using LLMs for knowledge graph construction. Insights were synthesized from the reviewed studies.

4.1.2 Eligibility Criteria

The study selection strategy was based on and adapted from the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) framework, which consists of four main phases: Identification, Screening, Eligibility, and Inclusion. The adoption of PRISMA ensured a standardized, transparent, and rigorous selection process [Page *et al*., 2021].

The research topic guided the formulation of the eligibility criteria. Only studies that met all inclusion criteria and none of the exclusion criteria proceeded to the final stage of data extraction and analysis.

4.1.3 Exclusion Criteria

- **EC1**: The article was not written in English.
- **EC2**: The article was not peer-reviewed (e.g., reports, editorials).
- **EC3**: The article was a secondary or tertiary study.

4.1.4 Inclusion Criteria

- **IC1**: The article addressed knowledge graph construction.
- **IC2**: The article focused on knowledge graph construction from textual data.
- **IC3**: The article proposed a method utilizing LLMs for knowledge graph construction from text.
- **IC4**: The article explained the technique or method developed.
- **IC5**: The article included an evaluation of the generated knowledge graphs.

4.1.5 Information Sources

The choice of databases was based on the following criteria: availability of a web-based search mechanism, inclusion of relevant sources, and support for title- and abstract-based searches. Based on these requirements, Scopus, IEEE Xplore, ScienceDirect, and ACM were selected due to their extensive collections of abstracts and citations from peer-reviewed sources.

4.1.6 Search Strategy

The search string was formulated based on the research topic and guiding questions. The term "Knowledge Graph" was used to refer to knowledge graphs. For Large Language Models, in addition to the expression "Large Language Model," synonyms such as "LLM" and "Generative Pre-trained Transformers" were included. These terms were combined using the *OR* operator. The keyword "text" was used to represent text-based data sources. The different terms were connected using the *AND* operator, resulting in the following search string:

"Knowledge Graph" AND ("Large Language Model" OR "LLM" OR "Generative Pre-trained Transformers") AND text.

The research topic guided both the search strategy and the eligibility criteria. The search was conducted in the selected databases using the query. The retrieved records underwent an initial screening based on bibliographic data, during which inclusion and exclusion criteria were applied. In the next phase, the selected studies' abstracts were thoroughly reviewed using the same criteria. Finally, studies that met all eligibility requirements proceeded to the data extraction stage.

4.1.7 Data Extraction and Synthesis Strategy

Data extraction was carried out in three stages: (i) compiling a reading form to organize information obtained from full-text reading; (ii) conducting a detailed analysis of concepts relevant to the research questions; and (iii) synthesizing the data for integration into the review.

The reading form included bibliographic details such as author, title, publisher, publication date, and country, as well as content-related information, including the general theme, problem or hypothesis, methodology, results, and key contributions. After completing the reading form, a comprehensive report was prepared to address the research questions and consolidate findings in the synthesis phase.

4.2 Review Conduction

This section outlines the execution of the state-of-the-art review and its results. It presents an account of the review process, detailing the data collection procedures.

Figure 1 illustrates the PRISMA flowchart, which represents the study selection process in a systematic review. The process is structured into three main phases: Identification, Screening, and Inclusion. The flowchart follows a linear sequence, depicting the selection process step by step. On the left, vertical labels identify each phase, while the boxes on the right outline the inclusion and exclusion criteria applied.

In the Identification phase, the search string was executed across the Scopus, IEEE Xplore, ScienceDirect, and ACM databases on March 31, 2025, yielding a total of 321 studies, of which 48 were duplicates.

An initial set of 273 studies was filtered based on bibliographic data and predefined exclusion criteria. Studies not written in English (EC1) led to the removal of 13 entries. The exclusion of non-peer-reviewed works, including reports, book chapters, editorials, and other non-peer-reviewed publications (EC2), accounted for 48 studies. Additionally, four studies were excluded because they were secondary or tertiary research (EC3). After applying these criteria, 208 studies remained for further assessment.

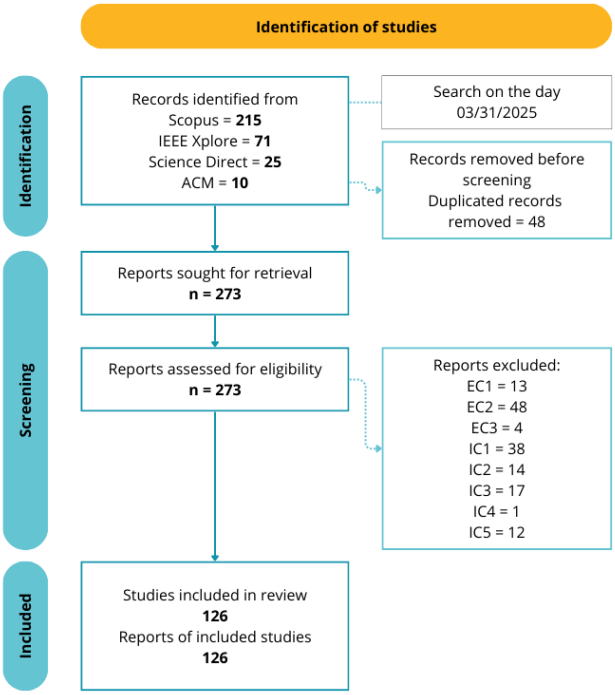

Figure 1. Review flowchart

The abstracts of the remaining 208 studies were reviewed in detail, with inclusion and exclusion criteria reapplied. At this stage, the number of included studies was further reduced to 140. Among these, 170 studies addressed knowledge graph construction (IC1), while 156 specifically focused on constructing knowledge graphs from textual data (IC2). Of these, 139 used LLMs for knowledge graph construction (IC3).

At the end of the screening phase, the full texts were read, and the remaining inclusion criteria were verified. Finally, 138 studies described the technique or method developed (IC4), and 126 studiesevaluatedf the generated knowledge graphs (IC5).

4.3 Reporting the Review

This section aims to present and analyze the research and to report and answer the research questions. The data supporting the analysis can be accessed through the Google Drive repository[1].

4.3.1 Which LLMs have been most commonly used for this purpose?

Among the 126 studies analyzed, 36 compared multiple models. As shown in Figure 2, which presents a pie chart of the models reported in the literature, the most frequently used LLM was OpenAI GPT, followed by LLaMA and ChatGLM.

OpenAI GPT models, commonly known as ChatGPT, have been reported in multiple versions. Figure 3 displays these versions in a bar chart, indicating that GPT-4 and its variants were the most frequently used.

[1] https://drive.google.com/drive/folders/1GVBULi4RgmpVqri2MreDfsoAfTEELe6A?usp=sharing

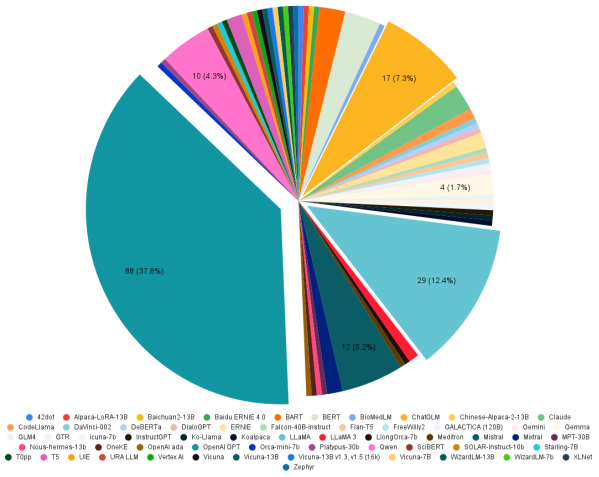

Figure 2. LLM frequency

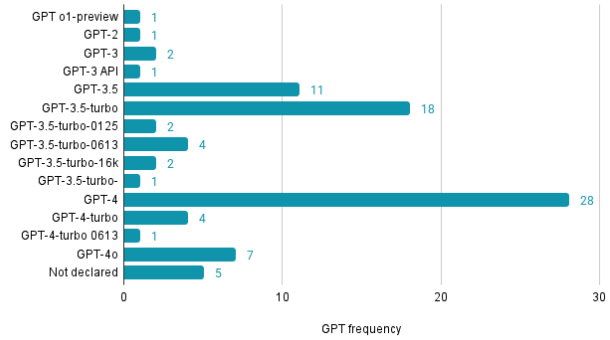

Figure 3. ChatGPT version

4.3.2 *In which domains have LLMs been most frequently applied for this purpose?*

The analyzed articles span multiple application domains. When specified, each study addresses one or more thematic areas.

Figure 4 presents a bar chart of the reported domains, indicating that general knowledge is the most frequently studied area, followed by health and manufacturing processes. Although knowledge graphs have broad applicability, current research is concentrated in these domains.

4.3.3 *What are the key advantages of using LLMs for this purpose?*

The reviewed studies identify three primary advantages of employing LLMs for knowledge graph construction from text: automation, scalability, and precision. Automation minimizes manual effort in extracting entities, relations, and triples. Scalability enables the processing of large datasets using adaptable frameworks across domains. Precision is demonstrated by improved entity and relation identification and the generation of more consistent triples.

4.3.4 *What are the main limitations of using LLMs for this purpose?*

Several studies report both advantages and limitations in overlapping areas, underscoring that the field remains in development and that no single approach comprehensively addresses all challenges.

Three recurring limitations are identified: manual curation, limited generalization, and low precision. Manual curation highlights the continued need for human intervention in system configuration, ontology development, and rule-based processing. Limited generalization refers to challenges in adapting techniques to new domains, often due to domain-specific dependencies. Low precision is linked to issues such as false positives, semantic ambiguity, and systemic errors, particularly in named entity recognition and triple generation.

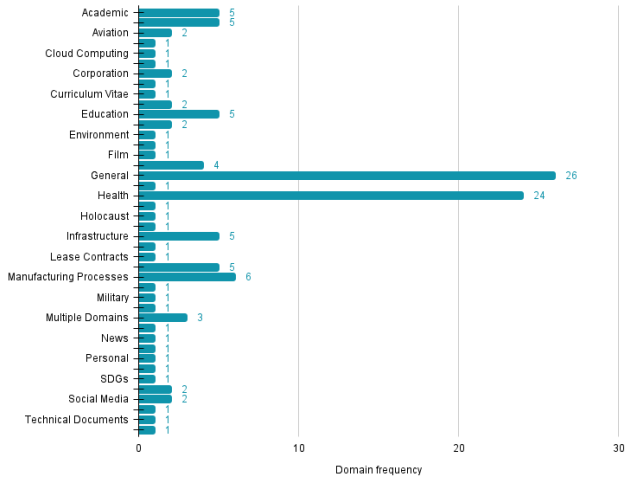

Figure 4. Domain versus publication count

5 Constructing KG from Text Source Using LLM

To understand the methods identified in this review, excerpts from the articles describing the techniques used in each study were extracted and analyzed.

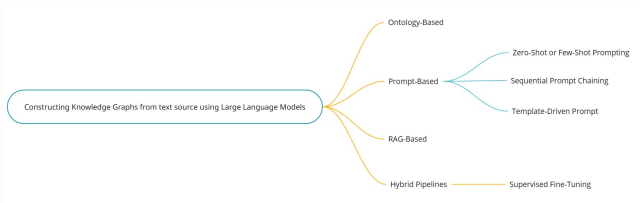

Figure 5. Categories and subcategories

The extracted excerpts were analyzed and grouped by similarity, as illustrated in Figure 5, resulting in four groups: Ontology-Based, Prompt-Based, RAG-Based, and Hybrid Pipelines. These groups are not mutually exclusive, meaning that a single study may employ more than one technique or may not fully fit into a single category.

5.1 Ontology-Based

Ontology-Based approaches constitute the most frequently identified category in the literature, encompassing approximately 47 primary studies. In this category, ontologies, schemas, or controlled vocabularies are explicitly adopted as the semantic backbone of the knowledge graph. These semantic artifacts are commonly formalized using standardized rules, such as the Resource Description Framework (RDF), Ontology Web Language (OWL), or related graph-based models. Overall, Ontology-Based approaches emphasize semantic alignment, conceptual consistency, and inter-

operability with external knowledge sources, often prioritizing correctness over full automation.

From a methodological standpoint, Ontology-Based pipelines typically begin with a careful delimitation of the target domain, followed by the selection, adaptation, or development of a domain ontology that captures the relevant concepts, relations, and constraints. Once this semantic structure is established, ontology-level information is injected into the extraction process. This may occur through explicit schema-driven extraction rules, ontology-aware named entity recognition, or ontology-informed prompts provided to Large Language Models. In such settings, LLMs are typically used as supporting components, assisting with entity recognition, relation extraction, or annotation, rather than serving as the sole mechanism for knowledge graph construction. Subsequent stages focus on mapping extracted elements to ontology classes and properties, followed by validation, instantiation, and storage of the resulting graph.

Queiroz *et al*. [2023] presents a canonical example of an ontology-centric pipeline, describing the development of an OWL 2 ontology for the domain of well engineering and its use for structuring information extracted from daily operational reports. The authors adopt a systematic ontology engineering process aligned with established standards such as BFO, IOF, and ISO 15926, thereby ensuring semantic rigor and interoperability. Textual inputs are processed using SpaCy for part-of-speech tagging and named entity recognition, complemented by regular-expression-based patterns tailored to domain-specific terminology. LLMs are used during the early stages of extraction. Extracted terms are explicitly mapped to ontology classes and object properties, and RDF triples are instantiated accordingly, as a main characteristic of ontology-based systems. The resulting knowledge graph is managed in Protégé and queried via SPARQL. Evaluation is conducted through competency questions and logical consistency checks using the Pellet reasoner. The authors report qualitative improvements in semantic coverage and queryability over unstructured text processing, while acknowledging the substantial manual effort required for ontology development and maintenance.

Cauter and Yakovets [2024] proposes a more extraction-oriented, yet ontology-grounded, approach that investigates ontology-guided knowledge graph construction from short maintenance texts. In this work, a lightweight ontology defining six permissible relations is embedded directly into structured prompts used for in-context learning. The LLM generates triplets constrained by the ontology, without producing RDF or OWL representations. Evaluation extends beyond traditional precision, recall, and F1 metrics to include ontology conformance and hallucination rates, explicitly measuring whether generated subjects, relations, and objects are consistent with both the input text and the ontology. The study demonstrates that ontologies can function as effective semantic constraints even in inference-only, prompt-based settings, reducing hallucinations and enabling semi-automated knowledge graph construction.

Ontology-based approaches are also represented in multimodal, reasoning-intensive pipelines, as illustrated by Shim *et al*. [2025]. The OmEGa framework targets manufacturing documents that combine text, tables, and images. It integrates OCR, table linearization, and layout embeddings with LLM-based extraction modules fine-tuned via Low-Rank Adaptation (LoRA), which adapts models by freezing original weights and training adapted matrices. Unlike lighter ontology-guided prompting approaches, OmEGa employs a domain ontology formalized in OWL and applies the Semantic Web Rule Language (SWRL), which combines OWL and Rule Markup Language to express inference and logic rules, to perform ontology-based reasoning over the extracted graph. The resulting knowledge graph is treated as an ABox [2] and enriched through inference, yielding substantial increases in the number of validated and inferred triples. Quantitative evaluation shows that ontology-based reasoning improves both precision and recall of the final graph compared to extraction without reasoning.

Mihindukulasooriya *et al*. [2023] provides a complementary perspective by introducing Text2KGBench, a benchmark specifically designed to evaluate ontology-driven knowledge graph generation from text. Rather than proposing a novel construction pipeline, this work focuses on standardizing datasets, ontologies, prompts, and evaluation metrics. Ontologies are expressed in OWL and verbalized for compatibility with LLM prompts, while outputs are generated as textual triplets. The benchmark defines multiple evaluation dimensions, including factual extraction accuracy, ontology conformance, and hallucination rates. By decoupling ontology-driven extraction from specific system architectures, Text2KGBench highlights the central role of ontologies as both guidance mechanisms and evaluation criteria.

Finally, some studies employ ontologies primarily as extraction-time knowledge without adopting Semantic Web representations in the final artifact. The AutoRD system proposed by Cao *et al*. [2024] integrates rare-disease ontologies such as ORDO, Mondo, and HOOM to enrich prompts and guide entity and relation extraction with GPT-4. Ontologies are used to improve recall and precision during extraction and calibration. However, the resulting knowledge graph is represented as triplets stored in a graph database, Neo4j, rather than serialized in RDF or OWL. The authors report quantitative improvements over base LLMs and conduct ablation studies demonstrating the contribution of ontology-enhanced prompting, exemplars, and prompt reminders, while also noting limitations related to dataset scope and residual classification errors.

Taken together, these studies illustrate distinct realizations of the Ontology-Based paradigm. At one end of the spectrum, works such as Queiroz *et al*. [2023] and Shim *et al*. [2025] adopt ontologies as first-class semantic artifacts, relying on formal OWL modeling, explicit instantiation, and logical reasoning to ensure semantic consistency and enable inference. At the other end, approaches such as Cauter and Yakovets [2024] and Cao *et al*. [2024] leverage ontologies primarily as constraints or enrichment mechanisms within LLM prompts, trading formal semantic representations for flexibility and reduced engineering overhead. Benchmark-oriented efforts such as Mihindukulasooriya *et al*. [2023] occupy an intermediate position, using ontologies to de-

[2] Assertional Box statements are the facts, data, or assertions about specific instances in a domain

fine evaluation standards rather than operational pipelines. Despite methodological diversity, all Ontology-Based approaches share a commitment to explicit semantic grounding and the continued relevance of ontological knowledge in LLM-assisted knowledge graph construction.

5.2 Prompt-Based

Prompt-Based approaches comprise 23 studies and rely on prompt engineering to extract structured knowledge from text without predefined ontologies or supervised fine-tuning. In this category, the LLM itself serves as the primary mechanism for identifying entities and relations, guided solely by carefully designed prompts.

The pipeline typically begins with a corpus of text, often composed of general-purpose documents, although some studies focus on specialized domains such as education or public policy. A prompt, designed as a zero-shot, few-shot, sequential, or template-based instruction, is then provided to an LLM, most commonly an OpenAI GPT model. The model outputs structured triples or triple-like representations, which may subsequently be cleaned and stored in a graph database or serialized in a lightweight RDF-like format.

The resulting graphs are usually instance-level and lack an explicit ontological layer. As a result, they prioritize rapid construction and adaptability over semantic rigor. This category is particularly useful in low-resource scenarios or when rapid domain adaptation is required.

Despite their flexibility, Prompt-Based approaches are susceptible to hallucinations, inconsistencies, and limited support for complex or multi-hop relations. Prompt design is a critical factor in controlling output quality; however, determinism cannot be guaranteed because LLMs are probabilistic.

This category is further divided into three subcategories: *Zero-Shot or Few-Shot Prompting*, *Sequential Prompt Chaining*, and *Template-Driven Prompting*.

5.2.1 Zero-Shot or Few-Shot Prompting

Zero-Shot or Few-Shot Prompting is particularly prevalent in technical documentation domains, where annotated resources are often scarce and rapid deployment is required. In this approach, the LLM is provided with a generic extraction instruction that may be optionally supplemented with a small number of illustrative examples. The model is expected to generalize the extraction task based solely on this prompt, without additional contextual grounding, external knowledge sources, or formal semantic constraints.

Methodologically, the pipeline feeds raw textual input directly to the LLM, which produces structured outputs following a predefined response format. As no external ontology or schema is enforced, the resulting knowledge graph typically consists of simple instance-level triples. Consequently, semantic control is primarily delegated to prompt design and to the LLM's internal representations.

Fang *et al.* [2024] constructs a knowledge graph from vehicle manuals using few-shot in-context learning. In this study, the input consists of raw explanatory and instructional text extracted from manuals. The LLM is prompted to automatically extract triples of the form $\langle subject, function, content \rangle$. After extraction, the subject and function fields are normalized using a unified vocabulary, and the resulting triples are stored in a Neo4j graph database. The constructed graph operates exclusively at the instance level and does not employ RDF, OWL, or RDFS.

The authors emphasize several advantages of this approach, including reduced human intervention, effectiveness in low-resource settings, and improved traceability for question-answering tasks. However, the study also reports limitations in overall accuracy, the restriction to simple one-hop queries, and difficulties in handling multimodal information, abbreviations, and semantic ambiguity.

5.2.2 Sequential Prompt Chaining

Sequential Prompt Chaining distinguishes itself by decomposing knowledge graph construction into multiple prompt-driven stages, each performing a specific, well-defined function. Rather than relying on a single prompt to extract all information at once, this approach distributes the task across successive steps, such as entity recognition, relation identification, validation, refinement, and aggregation, with intermediate outputs explicitly passed from one stage to the next.

From a methodological perspective, the pipeline typically begins by segmenting long textual inputs—such as documents, news articles, or reports—using a sliding-window strategy. This segmentation is followed by iterative prompting stages dedicated to named entity recognition, relation extraction between entity pairs, and relation prediction based on textual proximity. To further improve robustness, additional noise-control modules are incorporated, including semantic merging to fuse similar entities, group-based disambiguation supported by semantic checks or fact verification, and self-verification steps in which the LLM performs binary validation over previously extracted elements. Through this iterative and modular design, Sequential Prompt Chaining enables finer-grained control over the extraction process and generally achieves greater consistency than single-prompt approaches.

Geng *et al.* [2024] proposes a multi-task, iterative prompting framework explicitly designed for structured information extraction. In their approach, sliding-window text segmentation is combined with task-specific prompts for entity recognition and relation extraction, followed by semantic fusion and group disambiguation modules that consolidate overlapping or ambiguous entities. The pipeline further incorporates self-validation prompts to reduce noise and improve extraction reliability. The final output consists of a hierarchically organized set of subject–predicate–object triples, which are semantically refined but not formalized using RDF, OWL, or RDFS.

In terms of performance, the modular, iterative nature of the pipeline improves robustness and reduces noise, particularly compared to baseline extraction methods. The approach is evaluated on multiple benchmark datasets, including WikiANN, WikiNeural, ACE2005, CoNLL2003, CoNLL2004, and SciERC, achieving superior Micro-F1 scores across all tasks. Ablation studies further indicate that each module contributes incrementally to performance gains, with the iterative prompting component having the most significant impact [Geng *et al.*, 2024].

Despite these strengths, the study also highlights essen-

tial limitations. The absence of formal semantic representations limits interoperability and reuse, and the pipeline's overall effectiveness remains strongly dependent on prompt design and manual tuning. Moreover, residual hallucination effects persist [Geng *et al*., 2024].

5.2.3 Template-Driven Prompting

Template-Driven Prompting is the least frequent subcategory within Prompt-Based approaches. This strategy relies on rigid and highly structured prompt templates with predefined slots, such as fixed subject–predicate–object patterns, which are dynamically populated with input text. As a result, the extraction process is more constrained to improve consistency and reproducibility across runs, particularly in domains that require strict terminological control.

Methodologically, this approach typically combines LLM-based extraction with extensive manual review and normalization steps. In practice, the LLM generates candidate entities, relations, and attributes using a fixed template, while human experts validate, refine, and standardize the outputs. Consequently, Template-Driven Prompting often results in large-scale, domain-specific knowledge graphs, albeit with reduced automation.

Li *et al*. [2024a] describes the construction of a traditional Chinese medicine knowledge graph focused on rheumatism. The corpus consists of texts extracted from ancient medical books, classical works, and medical prescriptions. A three-part prompt structure is employed, comprising the medical text, explicit extraction instructions, and refinement instructions. Extraction is followed by a multistage manual refinement process involving validation against project standards and expert review. The resulting knowledge graph is represented as structured triples stored in Neo4j, supports domain-specific querying, and is publicly available. Although the use of LLMs reduces manual labeling effort, the pipeline remains dependent on expert validation and lacks automated semantic reasoning mechanisms.

5.3 RAG-Based

RAG-Based approaches encompass 18 primary studies and explicitly integrate Retrieval-Augmented Generation mechanisms into knowledge graph construction or enrichment pipelines. In this category, LLMs do not rely solely on their internal parametric knowledge; instead, generation is grounded in externally retrieved evidence. Compared to ontology-based approaches, RAG-based methods prioritize scalability and adaptability, while offering weaker guarantees of semantic consistency.

Methodologically, RAG-based pipelines typically begin with an incomplete triple, a graph-derived query, or a domain-specific information need. This input is used to retrieve relevant content from external corpora using dense retrieval strategies. Retrieved evidence is subsequently filtered or re-ranked and incorporated into the LLM prompt, enabling generation to be conditioned on explicit textual sources.

Wei and Esquivel [2024] presents a RAG-based framework for knowledge graph completion, in which incomplete triples are=$\langle h, r, ? \rangle$ are transformed into natural language queries. In this representation, h denotes the head entity (subject of the triple), r represents the relation (predicate), and ? indicates the missing tail entity to be predicted. Relevant passages are retrieved via dense semantic retrieval and provided to an LLM, which generates the missing element. The resulting triples are integrated into the knowledge graph without explicit RDF or OWL serialization. Evaluation on benchmark datasets demonstrates improved performance over embedding-based completion methods.

Tian *et al*. [2025] investigates the construction of a multimodal knowledge graph in the manufacturing domain using RAG-based and LLM-assisted techniques. The approach integrates textual, visual, and structural data sources, guided by domain ontologies and expert-defined constraints. Although formal Semantic Web standards are not employed, the method supports semantic search, question answering, and reasoning over manufacturing processes.

Overall, RAG-Based approaches ground generation in retrieved evidence, improving factuality and adaptability. However, reliance on retrieval mechanisms introduces variability across executions and limits reproducibility.

5.4 Hybrid Pipelines

Hybrid Pipelines comprise 37 studies and combine conventional information extraction components with LLM-based capabilities. In this category, classical NLP modules, such as named entity recognition and relation extraction, serve as the primary extraction mechanisms. At the same time, LLMs are used for refinement, normalization, validation, or semantic enrichment.

Hybrid pipelines typically begin with raw documents processed by supervised or rule-augmented models. Extracted entities and relations are transformed into triples and may undergo additional steps such as normalization, noise reduction, or cross-sentence aggregation. Compared to purely prompt-based approaches, Hybrid Pipelines generally report higher precision and lower hallucination rates, albeit at the cost of annotation effort and domain expertise.

5.4.1 Supervised Fine-Tuning

Supervised Fine-Tuning represents a dominant subcategory within Hybrid Pipelines. In this paradigm, transformer-based models are fine-tuned on annotated corpora to perform domain-specific extraction tasks.

Ji *et al*. [2024] describes the construction of a large-scale medical knowledge graph using a fine-tuned biomedical language model. Entities and relations are extracted from scientific abstracts, mapped to structured triples, and evaluated using precision, recall, and F1-score.

Li *et al*. [2024b] presents a biomedical extraction pipeline combining fine-tuned models and rule-based signals, incorporating novelty prediction to label relations. Evaluation demonstrates improved robustness compared to baseline approaches.

Gao *et al*. [2024] proposes a joint entity–relation extraction method for long biomedical texts using fine-tuned transformer models augmented with graph convolutional networks. Evaluation across multiple datasets shows improved extraction quality.

Ghanem and Cruz [2024] compares supervised fine-tuning and prompting strategies for knowledge graph construction, reporting superior performance for fine-tuned

models while highlighting evaluation challenges related to synonymy and hallucination.

6 Comparative Analysis

The analysis identified four distinct groups of LLM-based methods for building knowledge graphs, which differ in their level of semantic formalism, reliance on prompts, architectural complexity, and intended use cases. To support the selection of the most suitable method for building knowledge graphs based on LLMs, a comparative analysis was conducted. This analysis considers five criteria: exactness, scalability, domain adaptability, reproducibility, and ease of implementation.

Exactness refers to the ability of the method to generate correct and semantically coherent triples. Scalability evaluates performance as the volume of data grows. Domain adaptability refers to the ease with which an application can be applied in various contexts and areas of knowledge. Reproducibility refers to the consistency of results obtained when the same process is repeated multiple times. And finally, ease of implementation encompasses the demand for infrastructure, execution time, and computational costs.

The classification of each category into High, Medium, or Low levels was defined based on the frequency of the criteria in the studies analyzed. A High level was assigned when the majority of studies reported positive evidence for a given criterion, corresponding to 60%–100% of the evidence. A Medium level was applied when the findings indicated a balance between positive and negative evidence, ranging from 30% to 60%. Finally, a Low level was assigned when limitations or unsatisfactory results predominated, corresponding to 0%–30% of the evidence.

The Table 2 summarizes the results obtained and highlights relevant differences between the four categories analyzed.

Ontology-based approaches achieve consistently high precision and adaptability due to the explicit semantic constraints imposed by ontologies and schemas. Their medium scalability reflects the engineering effort required to extend ontologies and maintain consistency as data volume grows. Reproducibility is also moderate, as variations may arise from manual modeling decisions and ontology evolution over time.

Prompt-based approaches demonstrate high accuracy in controlled settings but perform only medium across all other criteria. Their reliance on prompt formulation and probabilistic generation introduces variability across executions, limiting reproducibility. While these approaches are attractive for rapid prototyping, their sensitivity to prompt design constrains their reliability in large-scale or long-term deployments.

RAG-Based methods combine high exactness and ease of implementation by grounding generation in retrieved evidence. However, reproducibility is low, as retrieval results may vary across runs, corpora updates, or indexing strategies. Scalability and adaptability are moderate, reflecting the dependence on retrieval infrastructure and the quality of external corpora.

Hybrid Pipelines achieve high exactness and adaptability by leveraging supervised or rule-based extraction as a stabilizing core, complemented by LLM-based refinement. Nevertheless, reproducibility remains limited due to the complexity of coordinating multiple components and the sensitivity of learning-based modules to training data and configuration choices.

Beyond qualitative categorization, the analysis indicates that each category aligns with different evaluation emphases. Ontology-Based methods are well-suited to evaluations based on ontology conformance, logical consistency, and competency questions. Prompt-Based approaches are commonly evaluated using triple-level precision, recall, and F1-score, as well as hallucination rates. RAG-based methods are frequently evaluated using downstream task performance metrics alongside factual consistency checks. Hybrid Pipelines tend to support the most comprehensive quantitative evaluations, including extraction metrics, ablation studies, and graph-level analyses.

Overall, the comparative analysis highlights that no single category dominates across all criteria. Instead, the choice of method should be guided by project-specific priorities, such as semantic rigor, scalability, reproducibility, or development cost.

7 Conclusion

This study investigated the use of Large Language Models to construct Knowledge Graphs from unstructured textual sources through a scoping review grounded in a transparent, systematic protocol. By analyzing 126 primary studies, the review mapped the current state of the art and organized the literature into four methodological categories: Ontology-Based, Prompt-Based, RAG-Based, and Hybrid Pipelines.

Throughout the article, the review progressed from foundational concepts to a structured examination of related surveys, followed by a detailed methodological description of the review process. The core contribution lies in synthesizing construction approaches and identifying recurring patterns in how LLMs are integrated into knowledge graph pipelines. The categorization clarifies the roles of ontologies, prompts, retrieval mechanisms, and supervised learning components, and highlights the trade-offs among semantic rigor, automation, scalability, and reproducibility.

The comparative analysis demonstrated that Ontology-Based approaches offer strong semantic guarantees and support logical reasoning, but require substantial manual effort and domain expertise. Prompt-Based methods enable rapid deployment and low entry barriers but suffer from variability and limited reproducibility. RAG-based approaches improve grounding and factuality by incorporating external evidence, but they introduce dependence on retrieval infrastructure and dynamic corpora. Hybrid Pipelines achieve high extraction accuracy and robustness by combining supervised components with LLM-based refinement, at the cost of increased architectural complexity and annotation requirements.

Despite significant advances, the reviewed literature reveals persistent limitations. Across categories, studies report challenges related to manual curation, limited generalization across domains, hallucination effects, and heterogeneous evaluation practices. The lack of standardized bench-

Table 2. Comparative Analysis of LLM-Based Knowledge Graph Construction Methods

Categories	Exactness	Scalability	Adaptability	Reproducibility	Ease of implementation
Ontology-Based	High (41/47)	Medium (26/47)	High (31/47)	Medium (18/47)	High (32/47)
Prompt-Based	High (21/24)	Medium (08/24)	Medium (16/24)	Medium (12/24)	Medium (16/24)
RAG-Based	High (15/18)	Medium (10/18)	Medium (11/18)	Low (3/18)	High (15/18)
Hybrid Pipelines	High (33/37)	Medium (20/37)	High (24/37)	Low (10/37)	High (26/37)

marks and consistent metrics complicates direct comparison between approaches and obscures the interpretation of reported performance gains.

From an evaluation perspective, the review indicates a growing need for more systematic and quantitative assessment frameworks. Future studies would benefit from combining multiple evaluation layers, including: (i) extraction-level metrics such as precision, recall, and F1-score; (ii) graph-level measures assessing structural completeness, redundancy, and connectivity; (iii) semantic evaluations based on ontology conformance, logical consistency, and competency questions; and (iv) task-oriented metrics derived from downstream applications, such as question answering accuracy or knowledge graph completion scores. Explicit reporting of hallucination rates, error types, and reproducibility across runs would further strengthen empirical comparisons.

As a limitation of this review, the rapid evolution of LLM architectures and tooling may have led to the exclusion of very recent studies not yet indexed in the consulted databases. Moreover, the analysis relied on evidence from the primary studies, which vary considerably in methodological rigor and depth of evaluation.

Future research should therefore focus on developing standardized evaluation protocols for LLM-generated Knowledge Graphs, designing benchmarks that span multiple domains and construction paradigms, and exploring hybrid evaluation strategies that integrate symbolic validation with statistical metrics. Advancing in these directions is essential for consolidating LLM-based knowledge graph construction as a mature and reliable research field.

Declarations

Authors' Contributions

JP and GB contributed to the conceptualization and design of this study. JP performed the validation and verification. GB is the primary contributor and writer of this manuscript. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Availability of research artifacts

The data and other materials created in this literature review are available at `https://drive.google.com/drive/folders/1XEwv4b-P7me9g08YrISU-QEhMAkU7n-w?usp=sharing`.

Use of Artificial Intelligence

Grammarly, version 14.1256.0, was used to check for grammar, punctuation, and spelling errors in this paper. ChatGPT, version 5, was used to verify the grammar, punctuation, translation, and spelling in this paper.

References

Alammar, J. and Grootendorst, M. (2024). *Hands-On Large Language Models*. O'Reilly Media. Book.

Berryman, J. and Ziegler, A. (2024). *Prompt Engineering for LLMs*. O'Reilly Media. Book.

Cao, L., Sun, J., Cross, A., *et al*. (2024). An automatic and end-to-end system for rare disease knowledge graph construction based on ontology-enhanced large language models: Development study. *JMIR Medical Informatics*, 12(1):e60665. DOI: 10.2196/60665.

Cauter, Z. and Yakovets, N. (2024). Ontology-guided knowledge graph construction from maintenance short texts. In *Proceedings of the 1st Workshop on Knowledge Graphs and Large Language Models (KaLLM 2024)*, pages 75–84. DOI: 10.18653/v1/2024.kallm-1.8.

Fang, Y., Chen, Y., Jiang, Z., Xiao, J., and Ge, Y. (2024). Effective and reliable domain-specific knowledge question answering. In *2024 IEEE International Conference on e-Business Engineering (ICEBE)*, pages 238–243. IEEE. DOI: 10.1109/icebe62490.2024.00044.

Gao, T., Zhai, X., Yang, C., Lv, L., and Wang, H. (2024). Joint extraction of entity and relation based on fine-tuning bert for long biomedical literatures. *Bioinformatics Advances*, 4(1):vbae194. DOI: 10.1093/bioadv/vbae194.

Geng, H., Shi, C., Jiang, X., Kong, Z., and Liu, S. (2024). An entity relation extraction framework based on large language model and multi-tasks iterative prompt engineering. In *2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 4763–4769. IEEE. DOI: 10.1109/smc54092.2024.10831494.

Ghanem, H. and Cruz, C. (2024). Fine-tuning vs. prompting: evaluating the knowledge graph construction with llms. In *3rd International Workshop on Knowledge Graph Generation from Text (Text2KG) Co-located with the Extended Semantic Web Conference (ESWC 2024)*, volume 3747, page 7. DOI: 10.3389/fdata.2025.1505877.

Ji, L., Du, S., Qiu, Y., Xu, H., and Guo, X. (2024). Constructing a medical domain functional knowledge graph with large language models. In *2024 IEEE 6th International Conference on Power, Intelligent Comput-*

ing and Systems (ICPICS), pages 491–496. IEEE. DOI: 10.1109/icpics62053.2024.10796042.

Jurafsky, D. and Martin, J. H. (2025). Speech and language processing. DOI: 10.4324/9780203461891-3.

Kejriwal, M., Knoblock, C. A., and Szekely, P. (2021). *Knowledge graphs: Fundamentals, techniques, and applications*. MIT Press. Book.

Li, D. and Xu, F. (2024). The deep integration of knowledge graphs and large language models: Advancements, challenges, and future directions. In *2024 IEEE 2nd International Conference on Sensors, Electronics and Computer Engineering (ICSECE)*. IEEE. DOI: 10.1109/ICSECE61636.2024.10729340.

Li, H., Xia, C., Hou, Y., Hu, S., Liu, Y., and Jiang, Q. (2024a). Tcmrd-kg: Design and development of a rheumatism knowledge graph based on ancient chinese literature. In *Proceedings of the 2024 IEEE International Conference on Medical Artificial Intelligence (MedAI)*, pages 588–593. IEEE. DOI: 10.1109/MedAI62885.2024.00083.

Li, Z., Wei, Q., Huang, L.-C., Li, J., Hu, Y., Chuang, Y.-S., He, J., Das, A., Keloth, V. K., Yang, Y., *et al*. (2024b). Ensemble pretrained language models to extract biomedical knowledge from literature. *Journal of the American Medical Informatics Association*, 31(9):1904–1911. DOI: 10.1093/jamia/ocae061.

Liang, X., Wang, Z., Li, M., and Yan, Z. (2024). A survey of llm-augmented knowledge graph construction and application in complex product design. *Procedia CIRP*, 128:870–875. DOI: 10.1016/j.procir.2024.07.069.

Mao, Y., He, J., and Chen, C. (2025). From prompts to templates: A systematic prompt template analysis for real-world llmapps. In *Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering (FSE Companion)*. ACM. DOI: 10.1145/3696630.3728533.

Mihindukulasooriya, N., Tiwari, S. M., Enguix, C. F., and Lata, K. (2023). Text2kgbench: A benchmark for ontology-driven knowledge graph generation from text. *arXiv preprint arXiv:2308.02357*. DOI: 10.1007/978-3-031-47243-5$_14$.

Muscolino, H., Machado, A., Vesset, D., and Rydning, J. (2023). Untapped value: What every executive needs to know about unstructured data. White paper, IDC, Framingham, MA. Available at:`https://images.g2crowd.com/uploads/attachment/file/1350731/IDC-Unstructured-Data-White-Paper.pdf`. Sponsored by Box.

Negro, A., Kus, V., Futia, G., and Montagna, F. (2022). *Knowledge Graphs and LLM in Action*. Manning. Book.

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., *et al*. (2021). The prisma 2020 statement: an updated guideline for reporting systematic reviews. *bmj*, 372. DOI: 10.1136/bmj.n71.

Pan, S., Luo, L., Wang, Y., Chen, C., Wang, J., and Wu, X. (2024). Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3580–3599. DOI: 10.1109/tkde.2024.3352100.

Peters, M. D. J., Godfrey, C. M., McInerney, P., Soares, C. B., Khalil, H., and Parker, D. (2015). The joanna briggs institute reviewers' manual 2015: methodology for jbi scoping reviews. Available at:`https://repositorio.usp.br/item/002775594`.

Phoenix, J. and Taylor, M. (2024). *Prompt Engineering for Generative AI*. O'Reilly Media. Book.

Queiroz, J., Jaculli, M. A., Junior, N. C., Silveira, I. M., Mendes, J. R. P., Penteado, B. E., Guilherme, I. R., and Perrout, S. R. (2023). An ontology of well engineering entities to extract and structure text data from daily reports. In *Proceedings of the 28th International Conference on Engineering Applications of Artificial Intelligence (EAAI)*, pages 1–8. Elsevier. Available as preprint. DOI: 10.2118/223801-ms.

Reynolds, L. and McDonell, K. (2021). Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*. DOI: 10.1145/3411763.3451760.

Schulhoff, S., Ilie, M., Balepur, N., *et al*. (2024). The prompt report: A systematic survey of prompting techniques. *arXiv preprint arXiv:2406.06608*. DOI: 10.48550/arXiv.2406.06608.

Shim, M., Choi, H., Koo, H., Um, K., Lee, K.-H., and Lee, S. (2025). Omega (ω): Ontology-based information extraction framework for constructing task-centric knowledge graph from manufacturing documents with large language model. *Advanced Engineering Informatics*, 64:103001. DOI: 10.1016/j.aei.2024.103001.

Tian, X., Xu, J., Wang, S., and Zhou, Y. (2025). Construction and application of a multi-modal knowledge graph integrated with large language models in the field of manufacturing processes. In *2025 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pages 0288–0292. IEEE. DOI: 10.1109/icaiic64266.2025.10920784.

Wang, Q., Li, C., Zhang, Y., Liu, Y., Xiong, H., and Shen, T. (2024). Knowledge graph construction: State-of-the art techniques and future opportunities. In *2024 12th International Conference on Information Systems and Computing Technology (ISC Tech)*. IEEE. DOI: 10.1109/ISCTech63666.2024.10845648.

Wei, Z. and Esquivel, J. A. (2024). Enhancing knowledge graph completion with retrieval-augmented generation using large language models. In *2024 6th International Conference on Frontier Technologies of Information and Computer (ICFTIC)*, pages 828–831. IEEE. DOI: 10.1109/icftic64248.2024.10912943.

Zhong, L., Wu, J., Li, Q., Peng, H., and Wu, X. (2023). A comprehensive survey on automatic knowledge graph construction. *ACM Computing Surveys*, 56(4):1–62. DOI: 10.1145/3618295.

Zhu, Y., Wang, X., Chen, J., Qiao, S., Ou, Y., Yao, Y., Deng, S., Chen, H., and Zhang, N. (2024). Llms for knowledge graph construction and reasoning: Recent capabilities and future opportunities. *World Wide Web*, 27(5):58. DOI: 10.1007/s11280-024-01297-w.

Zong, H., Wu, R., Cha, J., Feng, W., Wu, E., Li, J., Shao,

A., Tao, L., Li, Z., Tang, B., *et al*. (2024). Advancing chinese biomedical text mining with community challenges. *Journal of Biomedical Informatics*, page 104716. DOI: 10.1016/j.jbi.2024.104716.