External Knowledge Integration in Large Language Models: A Survey on Methods, Challenges, and Future Directions

Itisha Yadav [a,b,*], Sirko Schindler [a], Diana Peters [a] and Roman Klinger [b]
[a] *Institute of Data Science, German Aerospace Center (DLR), Jena, Germany*
E-mails: itisha.yadav@dlr.de, sirko.schindler@dlr.de, diana.peters@dlr.de
[b] *Fundamentals of Natural Language Processing, University of Bamberg, Germany*
E-mail: roman.klinger@uni-bamberg.de

Abstract. Large Language Models (LLMs) have shown effectiveness in various natural language understanding (NLU) tasks. However, they face notable limitations like hallucinations, a lack of contextual knowledge, and outdated or incomplete knowledge when applied across knowledge-intensive domains such as scientific research, biomedical sciences, finance, law, and others. These challenges commonly arise from the scarcity and under-representation of domain-specific data during the training and model alignment phases, the latter being synonymous with reinforcement learning from human feedback (RLHF). Furthermore, LLMs struggle to provide nuanced expertise, as their internal knowledge remains static and generalized, hindering their ability to reason accurately or deliver context-aware results in specialized tasks. This survey investigates the integration of external knowledge into LLMs to address these limitations. The focus is on decoder-based LLMs, i.e., autoregressive models that generate text sequentially. By investigating parametric and non-parametric approaches, this work discusses methods to enhance model reasoning capabilities, factual accuracy, and adaptability for domain-specific and knowledge-intensive tasks. Additionally, it highlights the potential of integrating external knowledge to improve explainability and ensure more trustworthy outputs. This survey supports software developers and natural language processing (NLP) researchers in designing natural language understanding systems for specialized domains by leveraging pre-trained LLMs. Additionally, the work provides a foundation for advancing LLM-based NLU systems with insights into future research areas.

Keywords: Large language models, Natural language understanding, External knowledge integration with LLMs, Retrieval augmented generation (RAG), Constrained-decoding with LLMs, Ontology-guided constrained-decoding with LLMs, Knowledge graph construction, Knowledge mechanisms in LLM

1. Introduction

Large Language Models (LLMs) have demonstrated strong performance across a range of natural language understanding (NLU) tasks. This is due to their ability to encode vast amounts of knowledge extracted from enormous data crawled from the Internet [49]. They acquire knowledge through the training phase, during which they process massive corpora of text data to learn statistical patterns and relationships

*Corresponding author. E-mail: itisha.yadav@dlr.de.

between words, phrases, and concepts. Unlike explicit knowledge repositories such as relational databases or structured knowledge bases, the knowledge in LLMs is encoded implicitly in their parameters. This implicit nature means that retrieving specific pieces of information is not straightforward and depends on probabilistic generation rather than deterministic querying [155]. Research shows that LLMs contain factual knowledge [49], but their knowledge is static, i.e, confined to the state of information at the time of training when used in isolation, or "as is". Relying solely on the knowledge embedded in these models' parameters presents several fundamental challenges [129], including hallucinations, outdated data, and a lack of domain-specific context. These challenges can be mitigated by external knowledge integration with LLMs.[1]

Tasks like sentiment analysis or spelling and grammar correction perform well with generic models because they rely primarily on linguistic pattern detection rather than deep subject understanding, and thus do not always require external knowledge integration. However, knowledge-intensive tasks such as information extraction, which involve identifying relevant entities and relationships from domain-specific documents to create triples for knowledge graph construction, require structured domain modeling and access to specific, context-rich details to produce accurate results. For instance, constructing knowlegde graph (KG) from scientific ontologies and domain-specific documents requires compliance with domain-imposed constraints on entities, relations, and validation rules, which must be effectively incorporated into the LLM's generation process [39, 155]. Moreover, in scientific fields, critical details are often stored in proprietary, confidential documents that are not included in publicly available datasets, including numerical measurements (e.g., temperature, pressure), material properties, procedural descriptions, and safety information which are derived from technical datasheets, experimental records, and internal reports. Handling these specialized documents requires precise contextual knowledge, encompassing domain-specific concepts, relationships, and constraints, which can be realized via integrating external knowledge with LLMs. In this work, we focus on integrating such externally available knowledge with LLMs to enhance performance on natural language understanding tasks, particularly knowledge-intensive ones such as information extraction (IE), knowledge graph construction (KGC), and knowledge graph population (KGP) for domains poorly represented in the generic dataset. The goal of this integration is to digitize and semantically structure complex textual data and semi-structured documents, thereby enabling downstream reasoning tasks. Beyond these, other knowledge-intensive NLU tasks, such as entity linking, question answering, fact verification, commonsense reasoning, scientific reasoning, and technical document summarization, also benefit substantially from external knowledge integration to ensure factual consistency and domain-specific accuracy.

Survey Purpose: There are several survey papers, as discussed below, that focus on model architecture, model benchmarking, hardware requirements, and explore various applications of LLMs. However, there is a noticeable shortage of literature offering an in-depth and focused discussion on approaches to knowledge integration. For instance, LLM families, such as GPT [123], have been compared, examining their internal architectures and datasets for training and fine-tuning, and their applications in diverse fields [68, 99]. Additionally, Wang et al. [162] shed light on the progression of language models, tracing their development from statistical and neural language models to transformers and, ultimately, LLMs. Another survey discusses open-source large language models, emphasizing aspects such as data collection, model architectures, and training methodologies [66]. The study by Dong et al. [32] focuses on "safeguards" and "guardrails" touching the ethical aspects of LLMs usage. Others investigate reinforcement learning using human feedback, fine-tuning on domain-specific datasets, and task-specific fine-tuning [94, 144] and transfer learning techniques [142]. Regarding the synergy between knowledge-based systems and LLMs, Yang et al. [139] provide a comprehensive examination of methods for integrating LLMs with knowledge-based systems. It provides an extensive overview of LLM evolution and architectural paradigms and discusses approaches for combining LLMs with knowledge resources such as knowledge bases, knowledge graphs,

[1] Other challenges associated with LLMs that are not directly mitigated by incorporating external knowledge include scalability limitations [177] and response inconsistency [130].

and retrieval-augmented generation (RAG) systems. However, it does not offer a detailed methodological analysis of the techniques used to integrate external knowledge into LLMs or address knowledge-intensive NLU tasks for domain-specific datasets. The study presents a broader perspective on what integration entails but lacks a systematic discussion of the underlying mechanisms that operationalize such integration. In essence, that work addresses the question, "how can LLMs be combined with knowledge-based systems in general?"

Building upon this foundation, the present survey extends the discussion from a conceptual overview to a methodological exploration. While prior studies, including Yang et al. [139], provide valuable insights into the general landscape of LLM-knowledge integration, there remains a lack of a comprehensive perspective that individually examines techniques for integrating external knowledge with LLMs for NLU-based systems, their challenges, and future applications. The current survey addresses this gap by examining methods and mechanisms for integrating external knowledge sources into LLMs to improve their performance on knowledge-intensive NLU tasks. It focuses on the question, "how can external, structured, or semi-structured knowledge be injected into LLMs to improve NLU in knowledge-intensive domains?" This not only enhances the domain-specificity of LLM responses but also addresses critical issues such as bias and misinformation, resulting in more reliable and trustworthy systems for users across various applications. By exploring these techniques, the paper aims to provide insights into improving model robustness and adaptability, thereby facilitating their adoption in real-world scenarios.

Intended Audience: The intended audience for this survey includes researchers, practitioners, and developers working in natural language processing (NLP), machine learning, the Semantic Web, and artificial intelligence who seek to enhance LLMs for knowledge-intensive tasks. These tasks include information extraction, question answering, knowledge graph construction, and knowledge graph population, particularly in specialized domains where external knowledge integration is necessary for accuracy and contextual understanding. By addressing the challenges and opportunities discussed in this paper, researchers can gain a deeper understanding of how to effectively leverage and integrate external knowledge to improve model accuracy, scalability, and applicability across diverse domains.

Scope and Literature Survey Methodology: The literature survey employed a taxonomy of search phrases and keywords, developed through iterative refinement during preliminary analysis. This analysis focused on foundational research in external knowledge representation, such as ontologies and knowledge bases. It also considered the limitations of LLMs in handling specialized scientific content. Additionally, approaches

Table 1
Overview of the selected literature corpus (2019–2025) grouped by different categorizations.

Category	Type	Count (%)	Examples / Comments
Paper Type	Survey / Review Papers	21 (11.4%)	Consolidation and meta-analysis studies
	Concept / Method Papers	151 (82.1%)	Theoretical or methodological contributions
	Dataset / Tool Papers	12 (6.5%)	Datasets or software tools required for implementation purposes
Publication Format	Conference Proceedings	70 (38.0%)	ACL, EMNLP, NeurIPS, ICML, AAAI
	Journal Articles	40 (22%)	IEEE, Applied Sciences, ACM Surveys
	arXiv Preprints Only	67 (37%)	Papers only available as preprints
	Tools / Others	7 (4.0%)	GitHub, Zenodo
Peer-Review Status	Peer-Reviewed	109 (59%)	Conference + Journal papers
	Preprints / ArXiv	68 (37%)	Not yet peer-reviewed
	Tools / Other	7 (4.0%)	Software and resources
Total Number:		**184 papers**	

for building domain-specific, knowledge-intensive NLU systems were examined, particularly those that combine formal ontologies with proprietary semi-structured documents to construct knowledge graphs for domains underrepresented in general LLM training data. The scope of this survey encompasses the following:

- An overview of the concept of LLM-based natural language understanding systems.
- Limitations of LLMs in knowledge-intensive tasks.
- Approaches for addressing these limitations through external knowledge integration, categorized into parametric and non-parametric methods.
- Future research directions for advancing knowledge-augmented LLM-based NLU systems.

To facilitate literature retrieval, a Python script was developed to query multiple academic databases using our taxonomy as input. The script systematically crawled scientific papers from open-source databases including Google Scholar[2], Semantic Scholar[3], DBLP[4], and ACL Anthology[5]. Papers from IEEE Xplore[6] and other sources with access restrictions were retrieved manually. The complete taxonomy, the Python script, and the crawling result are publicly available [166]. Research papers published between 2019 and 2024 were considered for inclusion, with a particular emphasis on those from 2022 to 2024 to ensure coverage of recent advances. Studies published in 2025 were manually selected based on their direct relevance to the survey's objectives. From the set of papers retrieved through the Python-based crawling script, relevant studies were manually selected for in-depth research by filtering them based on title or abstract. The Python script facilitated the retrieval of seeding papers, while pertinent additional studies were identified through manual searches. Consequently, a hybrid approach combining automated retrieval and manual searching was employed to collect the literature for the survey. Table 1 provides a quantitative overview of the selected literature corpus, summarizing paper types, publication formats, and peer-review status.

While every effort was made to ensure comprehensive coverage, certain methodological constraints should be acknowledged: First, the keyword-based retrieval process may have unintentionally excluded studies that address knowledge integration under different conceptualizations or terminologies. Second, the rapid pace of progress in LLM research implies that very recent publications, i.e., those appearing shortly before or after the completion of this survey, may not yet be indexed in the consulted databases. Third, the focus on English-language literature may have led to the omission of relevant work published in other languages. Notwithstanding these limitations, the resulting corpus is considered sufficiently broad and representative to capture the major trends and directions in contemporary research on LLM–knowledge integration.

This survey extends existing work by systematically examining knowledge integration in LLMs across both parametric and non-parametric paradigms, offering a more holistic view than prior studies that focus mainly on prompting and reasoning [139] or retrieval-augmented generation [39]. Kindly refer to Table 2 for an overview of the surveyed methods. Our distinct contribution lies in: (a) coverage of both parametric and non-parametric integration techniques, (b) emphasis on methodological aspects of knowledge-intensive NLU tasks, such as information extraction and knowledge graph construction, and (c) inclusion of recent developments in Retrieval-Augmented-Generation (RAG), knowledge editing, and constrained-decoding. Although this survey offers a methodological analysis of knowledge integration strategies, unified performance benchmarking across these diverse approaches remains difficult due to differences in evaluation tasks, datasets, and domain-specific requirements. Our primary focus is on the systematic categorization and in-depth analysis of methods, with performance considerations highlighted where methodologically needed.

[2]https://scholar.google.com/
[3]https://www.semanticscholar.org/
[4]https://dblp.org/
[5]https://aclanthology.org/
[6]https://ieeexplore.ieee.org/

2. LLM-based NLU Systems: An Overview

What are LLM-based NLU Systems? LLMs are large neural networks/deep learning models trained on vast corpora of text to capture patterns, semantics, and syntactic structures in natural language. Generally, models with hundreds of millions to several billion parameters are considered large. Examples, include BERT (340M parameters), GPT-3 (175B parameters), and T5 (11B parameters) [30, 123]. These models have demonstrated impressive capabilities in various NLP tasks, including sentiment analysis [65], text classification [37], machine translation [38, 150], named entity recognition (NER) [88], and others. Natural language understanding (NLU) is a subfield of natural language processing (NLP) that builds systems to comprehend and interpret human language. It bridges the gap between human language and machine comprehension [67]. It involves various tasks, including semantic role labeling, named entity recognition, relationship extraction, coreference resolution, intent detection, question answering, reasoning with textual information, and more. The NLU algorithms extract structured patterns from unstructured or semi-structured data[7] [138]. The emergence of language models such as BERT and the GPT series has led to significant improvements in NLU performance. Integrating LLMs with NLU systems has been shown to substantially improve the semantic understanding of natural language [53], thereby enhancing accuracy. Numerous implementations have been developed to facilitate this integration. For example, McTear et al. [95] have optimized their RASA-based [134] dialogue system by integrating the GPT-3.5-turbo model [173] to engage users in a motivational health coaching offering reflective dialogues. Rajasekharan et al. [125] combine LLM with Answer Set Programming (ASP) to do qualitative and mathematical reasoning. Mukanova et al. [101] propose a methodology that uses LLMs for performing the task of ontology enrichment and semantic processing. Other works also include: using LLMs for ontology engineering and knowledge graph creation [136]. While these examples illustrate key applications of LLMs in NLU, they do not represent an exhaustive list, as research in this area continues to evolve. However, despite their success, they still face limitations in dealing with domain-specific knowledge and ensuring factual correctness in generated responses [5, 14, 62, 103].

How do LLMs Perform NLU? A breakthrough came with the introduction of transformers by Vaswani et al. in 2017 [150]. Transformers, as originally defined, are encoder-decoder architectures that primarily rely on attention mechanisms. This mechanism models dependencies among elements of a sequence, such as words in a sentence in natural language processing or patches/pixels in an image in computer vision, and it assigns learnable weights that capture their relative relevance. Transformers serve as the fundamental architecture for LLMs, with different models leveraging specific components of the transformer structure. For instance, BERT utilizes only the encoder [30], whereas GPT is built solely on the decoder mechanism [123]. In contrast, models like BART incorporate both the encoder and decoder components [76]. LLMs perform NLU by learning contextual relationships between words and sentences. Our focus is on decoder-based models, i.e., the causal language models that produce the outputs autoregressively. These models are generative language models. In the training phase, they calculate the probability of a word's occurrence based on the previous context words, i.e., learn the statistical representation of language [15, 169]. In a fine-tuning phase, the pre-trained models are steered towards user-specific responses by applying task-specific fine-tuning or instruction fine-tuning. The work by OpenAI [123] offers valuable insights into the concepts of generative pre-training and discriminative fine-tuning. They demonstrate how large parameter neural networks, such as the GPT-family models, can offer a task-agnostic architecture for performing NLU tasks with better results than task-specific models. Furthermore, it has been observed that decoder-based GPT-type models excel in knowledge extraction compared to encoder-based models, except when the knowledge is a standalone word or composed of independent words [6]. LLMs reliance on large-scale static and generic text corpora can result in issues such as the generation of hallucinated facts

[7] In the context of this survey, we consider semi-structured data as documents like datasheets comprised of both tables with complex structures and floating text.

or incomplete reasoning, which poses challenges for tasks requiring factual accuracy or specialized knowledge [120]. Developing techniques that incorporate external knowledge sources for access to proprietary or domain-specific data could significantly mitigate these issues, enabling LLMs to deliver more accurate and contextually relevant outputs across various applications and domains [174].

3. Challenges of LLM-based NLU Systems

Since LLMs form the core of modern natural language understanding systems, any shortcomings in LLMs naturally propagate to the NLU systems built on top of them. In the following, we discuss typical limitations of LLMs that emerge when they are applied to domain-specific or knowledge-intensive tasks. These challenges directly affect LLM-based NLU systems, reducing their ability to reliably process, interpret, and reason over specialized information.

3.1. Hallucinations Leading to Limitations in Factual Accuracy

One of the primary limitations of LLM-based NLU systems is their tendency to generate grammatically correct but factually incorrect information, a phenomenon known as "hallucination" or "confabulation" [77]. Hallucination results in the generation of misinformation, making LLMs unreliable and ultimately reducing the accuracy of the output responses. In healthcare, e.g., LLM hallucinations can have serious consequences, such as providing medically incorrect guidance, which could lead to harmful patient outcomes [91]. In scientific fields, these inaccuracies may limit the effectiveness of LLMs in extracting knowledge and structuring data. One reason for hallucination is the lack of factual information from proprietary data [77]. When performing tasks such as information extraction, LLMs can produce extraneous or "hallucinated" content – for example, appending interpretations or qualitative judgments to a value extracted from a technical datasheet (e.g., stating that a measured temperature is "high" or "low"). Although the extracted key values may be accurate, LLMs often generate supplementary information that is not explicitly present in the source document, potentially introducing semantic noise or misinformation. Another prevalent form of hallucination involves numerical distortion – either by altering decimal values or generating entirely spurious numbers absent from the input source. Additionally, when processing technical datasheets (semi-structured documents), LLMs may generate misinformation as the generated information is neither explicitly nor implicitly present in the datasheets used for training. For instance, consider the following use-case. The prompt that produced the output below included a task instruction for *information extraction*, along with a machine-parsed version of the datasheet, originally in PDF format. The input structure is provided below for clarity[8]:

Prompt Structure:

```
Task: Ontology-based information extraction from technical datasheets

Instruction:
Extract relevant entities and relationships from the given technical datasheet.

Input Context:
[Parsed PDF Datasheet]
- Tables and floating-text representing experimental measurements.
[Domain Ontology]
- Associated domain ontology defining target concepts.
```

[8]Due to data privacy constraints, the original domain ontology concepts and datasheet contents are omitted from the example prompt.

```
Expected Output:
Structured information aligned with the specified ontology.
```

The model-generated output: *"There are apparent asymmetries in the bias applied to some of the samples (upper samples have a slight bias-rand), suggesting an inconsistency in the manufacturing process that could affect the quality of the final product."* In this output, the clause following "suggesting" constitutes a *hallucination*, as it introduces content not present in the source datasheet and thereby adds noise to the extracted information. Such hallucinations are particularly problematic for tasks such as *information extraction* and *knowledge graph construction*, where the accuracy and reliability of results depend on maintaining strict alignment with the underlying data sources. Relying on hallucinated data in these tasks can lead to inaccuracies and misrepresentations, ultimately affecting the reliability and applicability of LLM-generated knowledge. Therefore, understanding and investigating hallucinations in LLMs is important for their seamless application.

At a broader level, hallucinations are classified as intrinsic (resulting from internal parameters of the model) and extrinsic (due to integrated external knowledge) [24]. These categories can be further divided into subtypes, including fact-based hallucinations, where incorrect information is generated, and faithfulness hallucinations, wherein the output of the LLMs fails to align with the input prompt. Additionally, coherence hallucinations are characterized by the generation of incoherent text. Other types include relevance hallucinations, where the LLM produces an out-of-domain or irrelevant response, and sensibility hallucinations, which involve the generation of nonsensical text [51]. Researchers or practitioners of NLP have different perspectives on hallucinations. The survey by Cleti et al. [24] highlights the dual nature of hallucinations across diverse domains like healthcare/science and art/design. For fact-based fields like healthcare and scientific research, hallucinations lead to the generation of incorrect facts that are not acceptable. On the other hand, for philosophical and abstract fields like arts and design, hallucinations can foster creativity by generating unconventional or unseen outputs that inspire innovations. Therefore, the issue of hallucinations and ways of mitigating it depends on the respective domain, or at least the broader categorization of the domain.

One perspective on why hallucinations persist in LLMs is that existing guardrail mechanisms are inadequate and fail to effectively prevent them. According to Pantha et al. [111], generic guardrails face significant challenges when applied to specialized domains. Ideally, LLMs should rely on guardrails to issue alerts when they lack domain-specific understanding, signaling their inability to generate a reliable response. However, in reality, these mechanisms are often inadequate. The problem is that while some LLMs recognize their limitations and trigger alerts, many fail to detect their own uncertainty and instead generate misleading responses with unwarranted confidence. This failure stems from the lack of domain-specific understanding, preventing LLMs from correctly identifying when their outputs should be restricted. As a result, hallucinations persist. The inadequacy of generic guardrails highlights the need for domain-specific interventions. Research has explored customized guardrails designed to mitigate hallucinations in scientific applications. Proposed frameworks and methodologies address key challenges, including time sensitivity, contextualization of knowledge, and intellectual property concerns. By integrating domain-aware constraints, these solutions aim to enhance the trustworthiness and reliability of LLM-generated content in specialized fields.

3.2. Incompleteness and Outdated Data

LLMs struggle with processing or generating up-to-date information for the NLU task, as their training data is static and may not capture up-to-date domain-specific knowledge. This limitation arises because most pre-trained LLMs rely on datasets collected at a specific time, making them unable to reflect real-time changes, temporal trends, or evolving domain-specific knowledge [34, 39, 77]. For example, an LLM-based system used in the manufacturing sector for tasks such as reasoning, design assistance, or technical question-answering may miss recent innovations in materials, automation methods, or production standards [82]. This lack of timely updates can lead to outdated or incomplete insights, limiting the system's

usefulness in fast-moving industrial contexts. Similarly, in healthcare, where medical treatments, clinical guidelines, and biomedical discoveries evolve rapidly, an outdated LLM may generate recommendations based on outdated information, with potentially serious consequences. To overcome these challenges, it is crucial to integrate mechanisms that enable LLMs to access external, up-to-date knowledge bases or adapt dynamically to evolving datasets. The strategies for handling outdated knowledge enable models to bridge the gap between static training data and the real-time knowledge needed for accurate, reliable decision-making across diverse domains.

A common challenge with LLMs, closely related to the problem of incomplete or outdated knowledge, is their inconsistent response generation, that is, producing varied outputs for the same input. Even when the required external knowledge is available, the reliability of an NLU system also depends on the model's ability to generate stable and predictable outputs. Inconsistency arises from several factors, including limited or imbalanced training data, stochastic sampling strategies, sensitivity to prompt phrasing, training data biases, and architectural design choices [183]. Mathematically, LLMs are deterministic: given identical inputs and fixed model parameters (temperature = 0, fixed random seed), they produce the same output. However, most real-world applications intentionally introduce stochastic sampling through hyperparameters like temperature and top-p to promote diverse text generation. This causes LLMs to probabilistically sample tokens at each generation step, resulting in output variability across repeated prompts. Beyond stochasticity, another inherent source of inconsistency lies in the stateless nature of LLMs. Unlike systems such as recommender engines or chatbots that depend on historical user interactions or time-dependent profiles, LLMs operate without memory of previous exchanges. This statelessness improves privacy and security by avoiding storage of user-specific data, but it also presents difficulties for tasks requiring continuity or persistent reasoning. Without retained context, responses may appear arbitrary or disconnected when the input lacks explicit contextual cues. As a result, the stateless nature of LLMs contributes to the unpredictability of their outputs, leading to inconsistent responses [86, p. 7]. Such variability poses a significant concern, as it affects the reliability and trustworthiness of LLMs across a wide range of applications [130].

3.3. Lack of Contextual Understanding and Related Ambiguities

LLMs are sensitive to the noisy language of prompts. The lack of contextual understanding can contribute to linguistic ambiguities, as LLMs rely on statistical patterns inferred from the text rather than explicit reasoning [61]. For instance, words with multiple meanings, such as "bank" (which could refer to a financial institution or the side of a river), can be misinterpreted if the model lacks sufficient domain-specific knowledge or contextual clues. This limitation becomes particularly evident in specialized fields or nuanced conversations where precise understanding is critical. Consequently, semantic disambiguation in LLMs remains an active research area [170]. Keluskar et al. [61] analyze ambiguities in LLM-generated responses within the context of open-domain question answering. They argue that ambiguity primarily arises due to a lack of context, which includes not only textual cues but also social and psychological ones. Their study highlights the need to integrate external knowledge sources, such as knowledge graphs, to enhance clarity and disambiguation. Their results demonstrate that contextual enrichment, i.e., contextual information to LLMs, significantly reduces disambiguation and increases accuracy. Unlike scientific experts, LLMs struggle to understand the nuances of experimental setups or domain-specific methodologies. Humans can interpret the meaning of a document, identify and link related data across tables, and infer implicit relationships and contextual insights. LLMs, however, are generally unable to perform such reasoning reliably, limiting their ability to extract and contextualize knowledge from complex scientific texts. This often results in superficial or incomplete responses, such as misinterpreting parameters like temperature or pressure in experimental setups, because the context is not clearly defined.

Although LLMs possess relevant knowledge, they often struggle to apply it when prompts contain ambiguous entity types. Therefore, understanding different types of ambiguities is crucial for improving LLM-powered NLU tasks. Ambiguities can be categorized into three primary types: semantic, syntactic, and lexical [175]. Semantic ambiguity, also known as referential ambiguity, pertains to multiple interpretations of a word, phrase, or sentence. For example, the question, "What is the home stadium of the Cardinals?"

yields different answers depending on whether it refers to the Arizona Cardinals (football) or the St. Louis Cardinals (baseball) [61]. As this example illustrates, even humans struggle to resolve meaning without sufficient context, making it even more challenging for LLMs [175]. Lexical ambiguity arises at the word level and is related to parts-of-speech tagging. For instance, the word silver can function as a noun, adjective, or verb: "She bagged two silver (noun) medals.", "She made a silver (adjective) speech.", and "His worries had silvered (verb) his hair." [9]. Lastly, syntactic ambiguity concerns the grammar and structure of a sentence. A widely cited example is: "I saw the man with the telescope." This can be interpreted as either "I saw the man [who was holding the telescope]." or "I used the telescope to see the man." Different types of ambiguities contribute to vagueness, fuzziness, and uncertainty in language, leading to confusion in LLM responses. Augmenting contextual information or extending LLMs with external knowledge bases can mitigate these issues [137]. Additionally, LLMs struggle with self-verification, demonstrating a lack of self-consistency. This highlights a key challenge in polysemy resolution, emphasizing the need for further research into entity-type ambiguities and the broader complexities of language understanding [132].

4. Approaches of Integrating External Knowledge into LLM-based NLU Systems

In this survey, external knowledge refers to information supplied through domain-specific documents, structured resources like knowledge bases or relational databases, and ontologies that formally represent domain expertise. Such specialized knowledge covers factual details—concepts, relations, and constraints that are unique to a domain, but also extends to insights from unstructured sources, including narrative text or document passages. Both structured and unstructured knowledge play essential roles in enabling reasoning within a specific domain. For example, interpreting technical datasheets can greatly benefit from integrating domain-specific ontologies with a defined set of entities and relations to achieve a comprehensive understanding of the mentioned concepts. In specialized domains such as science and engineering, integrating external knowledge can reduce hallucinations, enhance reasoning, improve factual accuracy, and boost overall task performance. For instance, the work by Wang et al. [153] identifies challenges in using LLMs for manufacturing applications and recommends integrating an external knowledge base to generate industry-relevant insights. Another example of external knowledge integration can be seen in text-to-query systems, where LLMs translate natural language inputs into formal query languages such as SQL or SPARQL. In both cases, the model must understand and reason over the underlying structured schema—tables and relations in databases or entities and predicates in knowledge graphs, to generate syntactically correct and semantically meaningful queries. These systems demonstrate how LLMs integrate with structured external knowledge sources to produce actionable outputs, generating knowledge that extends beyond their pre-trained parameters.

Classification of knowledge: In the context of LLMs, data refers to text/corpora from books, websites, articles, among others, and knowledge is derived by processing the data during the training phase, i.e., the data is processed to create knowledge [27]. LLMs store internal knowledge within their parameters, which is derived during the training phase from large-scale, unstructured, and unlabeled datasets. For larger LLMs with over 100 billion parameters, this training data encompasses vast amounts of internet-scale information. The knowledge stored in model parameters is represented by numerical values known as weights, which are learned probabilistically through the backpropagation algorithm. This form of knowledge retention is referred to as *implicit memory* or *internal knowledge*. However, this knowledge is inherently limited as it only reflects data available up to the model's last training timestamp. Additionally, the training corpus is generic, often lacking specialized domain-specific information due to confidentiality restrictions that prevent such documents from being included in publicly available datasets. Furthermore, even within general datasets, highly detailed or technical documents may be disproportionately represented. Therefore, LLMs often requires external data to perform domain-specific or knowledge-intensive tasks, which serves as the additional contextual knowledge. Unlike implicit memory, this external data is maintained outside the model, typically in knowledge bases, relational databases, or structured file systems. When LLMs

Table 2
Overview of Surveyed Approaches

Approach	Related Literature
Parametric Knowledge	
Pre-training LLMs	BERT [30], GPT [123], MedBERT [149], LEGAL-BERT [17], FinBERT [112], TAPT [41], Knowledge Fusion Layer (K-XLNet) [168], Structured Knowledge-aware Pre-training [31], Learning Knowledge-Enhanced Representations [180], DKPLM [179], KE_PLM [50], Knowledge Representation Enhancement [8]
Fine-tuning LLMs	BioBERT [75], Knowledge Extraction and KG Construction [60], Knowledge-AI [102], Full Fine-tuning [30], LoRA [48], Prefix Tuning [79], Prompt Tuning [114], Bit-Fit [13], Adapter-based Fine-tuning [119], KG-Adapter [148], Adapter-based KG Integration [105], InfuserKI [152]
Knowledge Editing	MEND [116], EasyEdit [156], Knowledge Editing in LLMs [56], Knowledge Editing Survey [158], Editing for Bias Mitigation [55], ROME & MEMIT [96, 97]
Steering and Styling	sNeuron-TST [70], Tell [178], Focus [71]
Embedding / Graph Methods	Embeddings in LLMs [176], Graph Embeddings [57], KnowFormer [85]
Knowledge Distillation	DistilBERT [128], Knowledge Distillation in NeuralNets [43], MiniLM [159]
Non-Parametric Knowledge	
Prompting Methods	PEARL [143], Chain-of-Thought Prompting [164], ZEP [126], Knowledge Mechanism in LLMs 1-3 [6–8]
Knowledge Graphs / Ontologies	Introduction to KGs [36], KG + LLM Integration [174], Medical KG + LLM [163], GMeLLo [18], Knowledge Solver [35], Domain-specific KG Retrieval [160], KGQA with Planning-Retrieval-Reasoning [87], KG + LLM Reasoning [58, 90, 154], Scalability in KG + LLM Integration [177]
Constrained-Decoding	Lexically Constrained Decoding [44], Fast Constrained Decoding [118], Relation-Constrained Generation [20], JSON Mode [147], Ontologies for Constrained Decoding [89]
Memory-based Systems: Retrieval-Augmented-Generation (RAG)	RAG [77], RAG for LLMs [81], Elasticsearch [33], ChromaDB [23], Haystack [28], LangChain [93], Instruct Embeddings [141], Pre-training RAG Systems [39], Task-specific RAG Categorization [182], HALO: Medical QA RAG [10], Reliability RAG [54], PDF-based LLM-powered RAG [63], Extending Context Windows [19], Efficient Long-Context Generation [46], GraphRAG [113]
Other Memory-based Systems	Temporal Knowledge Graphs [126], MemGPT [108], MemOS [84]
Tool Usage / Function Calling	ToolFormer [131], LangGraph [72], LangChain [73], MeCo [78], Granite20B [1] Function Calling for KGs [42], Self-guided Function Calling [26], LLMs as Zero-shot Dialogue State Tracker through Function Calling [83], Function Calling with Generic LLMs [122]
Reinforcement Learning	Agents and Environment in RL [145], InstructGPT (RLHF) [106], RLKGF [167], RLAIF [12, 74], RL for Prompt Optimisation (RLPrompt) [29], StablePrompt [69]

consume and process such external information, it is referred to as *external knowledge*. Integrating external knowledge with LLMs is crucial for reducing hallucinations, enhancing contextual understanding, and ensuring access to up-to-date information. This integration can be done in two ways, at a broader level: (i) parametric knowledge integration, i.e., knowledge is encoded within the parameters of the LLMs, (ii) non-parametric knowledge integration, where the knowledge is not encoded within the model weights, but supplied from a storage system outside the LLM, like relational or knowledge databases. Table 2 summarizes the parametric and non-parametric methods, and the following sections explain these methods in detail.

4.1. Parametric Knowledge

As mentioned earlier, parametric knowledge is implicitly embedded within the model's architecture [6]. Therefore, parametric methods for integrating knowledge with LLMs involve modifying their internal

parameters and encoding/embedding information during training or fine-tuning.

4.1.1. Pre-training LLMs

The development of large language models (LLMs) typically follows a hierarchical training continuum that transitions from general-purpose learning to task- and fact-specific adaptation. At the foundation lies pre-training, where models are trained from scratch on large, unlabeled, and diverse corpora using self-supervised objectives (e.g., masked or causal language modeling) to acquire general linguistic and world knowledge. Foundational examples include encoder-based models such as BERT [30] and decoder-based models such as GPT [123]. Following the general pre-training phase, domain-adaptive pre-training (DAPT) continues training a pre-trained model on unlabeled, domain-specific corpora to better align its internal representations with specialized knowledge (e.g., biomedical, legal, or financial domains) [17, 112, 149]. For example, LEGAL-BERT [17] and FinBERT [112] are further trained on unstructured textual data from their respective domains—LEGAL-BERT on diverse English legal texts such as legislation, court cases, and contracts, and FinBERT on the Reuters TRC2 financial news corpus [104]. Both models demonstrate superior performance compared to the original BERT model on downstream tasks within their domains. In continuation of DAPT, task-adaptive pre-training (TAPT) further refines the model using unlabeled text drawn directly from the downstream task domain, enabling better alignment with the stylistic and distributional characteristics of the task data. This approach, formalized by Gururangan et al. [41], has been shown to improve model robustness and task-specific generalization before supervised fine-tuning.

Building upon these stages, several studies have explored enhancing the model by incorporating external structured knowledge into the language modeling process. For instance, Yan et al. [168] introduce a knowledge fusion layer on top of the transformer architecture to integrate knowledge graph information during pre-training without altering the underlying model structure. Similarly, Dong et al. [31] propose a structured knowledge–aware pre-training framework that embeds structured knowledge into the model using the masked language modeling (MLM) objective from BERT, enabling it to learn representations of complex subgraphs for improved performance on Knowledge Base Question Answering (KBQA) tasks. Related approaches that inject knowledge into the pre-training, DAPT, and TAPT phases include [50, 179, 180], all of which demonstrate that structured knowledge integration enhances model understanding and reasoning capabilities.

In the context of (large) language models, natural language understanding can be viewed as a process of manipulating and extracting knowledge, where relevant information is identified and adapted to meet task-specific requirements. One illustrative example is information extraction, which involves deriving structured signals, such as entities or relations, from data, retrieving pertinent facts encoded within the model, or classifying sentences into predefined categories [8]. Such operations highlight how LLMs leverage and transform knowledge to support a broad range of NLU tasks. However, despite their ability to store vast amounts of information, LLMs often struggle to extract and manipulate knowledge effectively. For instance, a model trained on the fact "Abraham Lincoln was born in Hodgenville, K.Y." may correctly answer the direct question "Where was Abraham Lincoln born?" but fail to respond to the inverse query "Who was born in Hodgenville, K.Y.?" unless explicitly trained with bidirectional mappings. This type of retrieval is called reverse retrieval. This limitation highlights that memorizing knowledge alone does not guarantee effective knowledge extraction. To address this gap, Allen-Zhu et al. [8] propose strategies to enhance LLMs performance by refining knowledge representation processes. One approach involves rewriting pre-training data using small auxiliary models to generate diverse permutations of knowledge, thereby improving retrieval flexibility. Another method focuses on incorporating fine-tuning data during pre-training, enhancing the model's ability to reason over stored information.

4.1.2. Fine-tuning LLMs

Unlike pre-training, DAPT, and TAPT, fine-tuning uses labeled, task-specific datasets and supervised learning objectives to adapt the model to specific applications, such as question answering, text classification, and summarization. The BERT paper [30] distinguishes pre-training and fine-tuning for larger language models. During fine-tuning, the parameters of a pre-trained language model are updated using labeled examples to optimize its performance on downstream NLP tasks [47, 140]. For instance, Jinhyuk

et al. [75] first adapt BERT to the biomedical domain through additional pre-training on biomedical text, resulting in BioBERT, and subsequently fine-tune it on labeled, task-specific datasets for downstream NLP tasks such as named entity recognition (NER), relation extraction, and question answering. LLMs, when fine-tuned, can be used to extract factual knowledge. Kazemi et al. [60] fine-tune LLMs to do knowledge extraction and knowledge graph construction. In general, fine-tuning has been effective in overcoming performance limitations on domain-specific NLU. The work of Muralidharan et al. [102] examines the effectiveness of LLMs in understanding and extracting information in the scientific domain. They propose a methodology called Knowledge-AI, which fine-tunes LLMs for downstream NLU tasks, such as question answering, NER, summarization, and text generation.

Exploring different approaches to fine-tuning LLMs is essential for understanding their practical implications, scalability, and resource requirements. Fine-tuning strategies differ primarily in the number of parameters updated during training, influencing computational efficiency, memory consumption, and model adaptability. **Full fine-tuning** modifies all parameters of an LLM, typically achieving strong task-specific adaptation. However, this approach is computationally expensive, requiring high-end GPU clusters, particularly for models with more than 100 billion parameters [124]. To address these constraints, the NLP community has increasingly turned to the **Parameter-Efficient Fine-Tuning** (PEFT) paradigm, which updates only a small fraction of parameters, or introduces a small number of additional trainable parameters—while keeping the core model weights frozen. This reduces hardware requirements and accelerates training, making fine-tuning feasible on modest computational setups. Within this paradigm, several techniques have emerged, including *Low-Rank Adaptation (LoRA) [48], prefix-tuning [79], prompt-tuning [114], BitFit [13], and adapter-based fine-tuning [119].* Among these, adapter-based methods are one of the earliest and most prominent forms of PEFT. These methods introduce small, trainable neural modules, known as adapters, within the transformer layers of a pre-trained model. During fine-tuning, only the adapter parameters are updated.

For instance, Shiyu et al. [148] combine knowledge graphs (KGs) with LLMs at the parameter level to improve reasoning. They note that simply adding KG information via prompting can introduce inconsistencies and lead the model to overly rely on its prior knowledge. To address this, they introduce a KG-adapter that leverages PEFT-based fine-tuning to embed both node and relation representations from the KG into the model. This approach enhances reasoning performance on KGQA. Similarly, Omeliyanenko et al. [105] introduce an adapter-based architecture for integrating KG knowledge into LLMs for link prediction tasks. Their approach inserts lightweight, trainable layers between the transformer blocks of a pre-trained model, enabling task-specific adaptation without full model retraining [157]. Extending these foundations, Wang et al. [152] propose an infuser-guided adapter integration framework—an optimized variant of adapter-based fine-tuning. This approach effectively mitigates catastrophic forgetting during knowledge integration. To ensure the added modules do not interfere with the model's internal representations, they introduce infuser-based adapters, which selectively inject knowledge only when the model lacks it. This selective infusion mechanism delivers superior performance compared to conventional adapter-based methods, indicating a promising direction for adaptive, knowledge-aware LLM fine-tuning.

Fine-tuning LLMs requires significant resources and computational power and comes with its own set of challenges. Kazemi et al. [60] highlight a downside known as "Frequency Shock", where fine-tuned models tend to overpredict rare entities while underpredicting common ones in the training data, ultimately degrading performance. Ghosal et al. [40] address another issue: "attention imbalance", where the attention mechanism unevenly prioritizes specific tokens over others. Ovadia et al. [107] compare two approaches for embedding factual knowledge into LLMs—unsupervised fine-tuning and Retrieval-Augmented-Generation (RAG). Their findings suggest that RAG outperforms unsupervised fine-tuning. They also note that unsupervised training often exposes models to multiple variations of the same fact, complicating the accurate retention of factual information. These challenges emphasize the importance of exploring non-parametric methods for integrating external knowledge into LLMs (cf. Subsection 4.2).

4.1.3. Knowledge Editing

Incorporating new knowledge into the parameters of LLMs through fine-tuning is computationally expensive due to their vast scale and the billions of parameters they contain. This challenge has motivated the development of more efficient mechanisms for integrating external knowledge without the high computational cost of full model retraining. Knowledge-based model editing (KME), or knowledge editing, addresses this by modifying only a small and targeted subset of parameters responsible for encoding specific information, thereby updating the model's knowledge while preserving its pre-trained capabilities [116, 156].

Unlike fine-tuning, which typically adjusts all or a broad range of parameters to optimize model performance for a downstream task, KME focuses on localized and semantically precise modifications. As highlighted by Ishigaki et al. [56], knowledge editing generally proceeds in two stages: (1) localization, where the model identifies the neurons or attention heads associated with the target knowledge, and (2) modification, where only those parameters are updated to reflect new or corrected information. This targeted approach enables locality, ensuring that updates affect only the intended knowledge, and generality, allowing the edit to generalize across semantically related contexts [158].

Beyond its application in LLMs, knowledge editing techniques have broader implications in machine learning, such as mitigating data biases and improving model robustness on downstream tasks [55]. By maintaining the integrity of previously learned information while efficiently integrating new knowledge, KME provides a computationally efficient and semantically stable alternative to fine-tuning, making it especially valuable for dynamically updating LLMs in knowledge-intensive domains.

To better illustrate the conceptual and computational distinctions between pre-training, fine-tuning, and knowledge editing, Table 3 provides a structured comparison highlighting their objectives, data requirements, training scope, efficiency, and key representative techniques. This comparative overview clarifies the boundary between full-scale model adaptation and targeted knowledge modification, situating knowledge editing as a lightweight yet powerful alternative for dynamically updating LLMs in knowledge-intensive settings.

4.1.4. Steering and Styling LLMs

Steering an LLM involves guiding its output to align with specific stylistic, instructional, or knowledge-driven constraints [70]. While it is commonly applied to Text Style Transfer (TST) – for instance, adapting an LLM to generate Shakespearean-style language, steering also plays a crucial role in knowledge integration, ensuring that models correctly interpret and apply external information in their responses. There are multiple perspectives on steering and styling LLMs, many of which facilitate the integration of external knowledge. Steering techniques can be broadly classified into parametric and non-parametric approaches[9]. Parametric methods, such as neuron deactivation or fine-tuning, modify model parameters to influence responses. On the other hand, non-parametric methods, such as prompt-based steering, control model behavior without altering the underlying parameters. Lai et al. [70] propose a novel parametric steering approach called sNeuron-TST for steering the style. This method identifies the neurons associated with source and target styles, deactivating the source-style neurons to enforce the target style in generated text. However, they observe that deactivating these neurons leads to performance degradation. To address this, they introduce a constructive decoding method that compensates for the removed neurons, improving output consistency. Beyond stylistic control, the similar parameter-level steering mechanisms of LLMs are also applicable to knowledge integration, as they direct the model's attention mechanisms to prioritize user-specified information (e.g., instructions or domain-specific knowledge). This is achieved by identifying subsets of attention heads and applying attention reweighting, which enables the model to effectively incorporate and process new information. Such steering methods have been shown to enhance performance on knowledge-intensive tasks [71, 178]. Therefore, steering techniques ensure that models remain aligned with the external knowledge sources.

[9] Approaches discussed in the context of knowledge-intensive tasks are mostly parametric and are thus discussed as parametric knowledge integration.

Table 3
Comparison of Pre-training, Fine-tuning, and Knowledge Editing in Large Language Models (LLMs).

Aspect	Pre-training (Sec. 4.1.1)	Fine-tuning (Sec. 4.1.2)	Knowledge Editing (Sec. 4.1.3)
Objective	Learn general linguistic and world knowledge from large unlabeled corpora.	Adapt pre-trained models for specific downstream tasks using labeled data.	Modify specific factual or conceptual knowledge without full retraining.
Data Type	Large-scale, unlabeled, diverse textual corpora.	Task-specific, labeled datasets.	Targeted factual or conceptual updates (e.g., correcting or adding specific facts).
Training Scope	Full model training from scratch (all parameters).	Updates all or a small subset of parameters or adds trainable modules while keeping the rest frozen (e.g., PEFT, adapters, LoRA).	Modifies only a small, localized subset of parameters or neurons [156].
Supervision Type	Self-supervised learning (e.g., MLM, CLM).	Supervised or semi-supervised learning.	Usually unsupervised or weakly supervised; guided by target knowledge specification [158].
Computational Cost	Extremely high (training from scratch with massive resources).	Moderate to high (depending on the number of parameters updated).	Low (localized parameter modification, no full retraining) [116].
Parameter Efficiency	Low, all parameters trained.	Moderate to high, PEFT methods improve efficiency.	Very high, edits only a few targeted parameters.
Knowledge Integration Mechanism	Implicitly learns knowledge from text corpora.	Injects task- or domain-specific knowledge via fine-tuning or adapters.	Identifies and modifies neurons encoding specific knowledge (localization + modification) [158].
Key Techniques / Examples	BERT [30], GPT [123], DAPT [17, 112], TAPT [41].	Full fine-tuning [30], LoRA [48], Adapters [119], Prefix/Prompt Tuning [79, 114]	MEND [100], ROME [96], MEMIT [97], EasyEdit [156].
Advantages	Builds strong general-purpose representations.	Enables domain/task specialization and improved downstream performance.	Efficiently updates or corrects model knowledge while preserving prior learning [55].
Limitations	High computational cost and data requirements.	Risk of catastrophic forgetting and resource-intensive for large models.	Limited scope of edits; potential instability in compositional reasoning [158].

4.1.5. Other Methods and Limitations of Parametric Approaches

Other methods of parametric data augmentation to LLMs include embedding techniques. They transform external knowledge into vector representations that can be integrated into the LLM's latent space, thereby capturing the text's linguistic features. The NLP community has used embeddings for years to transform raw textual information into numerical representations that can be processed by AI algorithms [176]. Additionally, methods such as graph embeddings [57, 110] and knowledge graph transformers [85] enable the model to learn richer representations of external knowledge, thereby improving its contextual understanding and reasoning abilities. Another prominent parametric technique is *knowledge distillation*, where a large teacher model transfers its knowledge to a smaller student model by training the student to approximate the teacher's outputs. This process embeds the teacher's knowledge directly into the student's parameters, allowing for model compression and efficiency gains while retaining much of the teacher's performance [43, 128, 159]. In the context of LLMs, knowledge distillation is widely used to reduce computational overhead and memory requirements, thereby making LLMs easier to deploy and more scalable.

Parametric knowledge within LLMs often faces challenges, such as a lack of explainability, as it is stored within the model weights; i.e., the knowledge is converted into numerical values, making its provenance difficult to trace. This might lead to security risks due to its opaque nature. Another challenge is that updating knowledge is computationally expensive and time-consuming, as parametric knowledge updates require some level of modification to model weights/layers/architecture. To address these limitations,

external non-parametric knowledge emerges as a good option, offering enhanced transparency, flexibility, adaptability, and operational simplicity [155]. Table 4 summarizes how these methods differ from non-parametric strategies in terms of computation, explainability, flexibility, and update mechanisms.

4.2. Non-Parametric Knowledge

As previously discussed, non-parametric knowledge refers to information or knowledge provided to the LLM without altering its internal weights or architecture. This type of knowledge is stored in a separate system outside the model and is not encoded within its trainable parameters. The following are the most commonly used approaches for integrating external and up-to-date knowledge into LLMs in a non-parametric manner.

4.2.1. Prompting Methods

Prompting refers to the process of providing textual instructions to an LLM to elicit desired task-specific behavior. Common prompting strategies include zero-shot prompting, where only the instruction is provided without examples; few-shot prompting, which supplements the instruction with a few input–output examples; and chain-of-thought (CoT) prompting, which guides the model by decomposing the reasoning process into intermediate steps [16, 143]. As a non-parametric approach, prompting does not alter the model's internal parameters or weights. Instead, it conditions the model externally via contextual cues by embedding relevant information directly into the prompt, within the LLM's context limit. While techniques such as chain-of-thought (CoT) prompting [164] improve reasoning tasks like retrieval and classification, they remain insufficient for more complex operations, such as inverse search, where models must infer relationships beyond explicitly stated data, as explained in Section 4.1.1. In such cases, advanced methodologies like retrieval-augmented generation (RAG) and reversal training are essential, as only a limited amount of context can be included in the prompt due to the LLMs 's context limits. Lack of context results in issues like hallucination and inappropriate response generation. To solve this problem, a new field of research emerged called context engineering [117, 126], in which relevant context is provided to the LLM to support knowledge-intensive NLU tasks. Retrieval-Augmented Generation (RAG) and Knowledge Graph (KG) integration, discussed in detail in the following sub-sections, can indeed be conceptualized as forms of context engineering. These findings highlight the importance of knowledge integration in LLMs, bridging the gap between limited contextual understanding and the effective application of knowledge in real-world, knowledge-intensive tasks [6–8]

4.2.2. Knowledge-based Methods: Knowledge Graphs and Ontologies

Knowledge Graphs and their role in LLMs: Knowledge graphs, such as DBpedia [11], or Wikidata [151], are external knowledge sources that store knowledge in the form of nodes and edges [36]. By linking LLMs with these graphs, it is possible to enhance the models' ability to reason about entities and their relationships. In recent years, significant research has focused on integrating knowledge graphs and LLMs to reduce hallucinations and improve factual accuracy. LLMs and KGs complement each other's capabilities. Merging both helps overcome each other's limitations. LLMs gain access to factual knowledge from KGs, which improves accuracy and trustworthiness, as in fact-checking. KGs, on the other hand, benefit from LLMs in language processing and language understanding tasks like: synthesis of user responses, question-answering, automated KG construction, etc. [22, 87, 110]. There are various ways to integrate KGs and LLMs. First, *KG-enhanced LLMs* leverage KGs as external knowledge sources to provide domain-specific context to LLMs, thereby improving the factual accuracy of their outputs. Second, *LLM-augmented KGs*, where LLMs are used for populating the triples in the knowledge graph, assist in KG-related tasks, and incorporate ontologies to ensure that the generated triples conform to domain-specific rules and constraints. Third is the *synergized LLMs+KGs*, a unified framework that aims to enhance each other's capabilities through knowledge representation and reasoning [110].

KG-enhanced LLMs, LLM-augmented KGs, and synergized KG+LLMs are methods for integrating KGs and LLMs. However, the focus of the work is KG-enhanced LLMs, where KG serves as the external source. For instance, Ye et al. [174] introduce a method to integrate KGs with Large Language Models (LLMs) to

enhance factual accuracy. Their approach employs deep reinforcement learning, which identifies relevant inference paths within the KG based on user input and incorporates this information into the LLM prompt, thereby providing context that yields more domain-specific and accurate results. The work by Wang et al. [163] discusses the challenges of handling evolving knowledge in the medical domain, emphasizing the need for continuous model updates and the integration of external knowledge. They propose a 3-step framework to develop an LLM-powered AI application for the medical domain: (i) modeling (which breaks down a complex task into simpler sub-tasks), (ii) optimization (enhancing the model response generation relevancy and accuracy by integrating external knowledge), and (iii) system engineering. They also highlight that more research is needed on system optimization, specifically on augmenting external knowledge with LLMs. Chen et al. [18] handle out-of-date information issues in LLMs by integrating knowledge graphs to facilitate accurate fact identification and logical reasoning. They propose a method called Graph Memory-based Editing for Large Language Models (GMeLLo), which combines the strengths of both KGs and LLMs to perform multi-hop question answering in dynamic environments, i.e., those with frequently updated external knowledge. Such advancements not only enhance the performance of LLMs in natural language understanding tasks but also foster greater trust and adoption of these models across critical fields such as healthcare, science, and education. Building on these integration strategies, researchers have explored other practical applications that leverage the synergy between KGs and LLMs to address specific tasks, such as question answering and improving explainability. Feng et al. [35], integrate KGs with LLMs to develop a multiple options question answering system. They call their methodology a knowledge solver, which allows them to search for relevant facts in the integrated knowledge graph. The approach also increases the explainability of LLMs' reasoning processes by providing complete retrieval paths, as demonstrated through experiments on the datasets: CommonsenseQA [146], OpenbookQA [98], and MedQA-USMLE [59]. There are other works in the same research area. For instance, Wang et al. [160] highlight the importance of connecting domain-specific KGs to LLMs for the task of domain-related question answering. They propose a subgraph retrieval method based on the chain-of-thought and PageRank, which returns the paths most likely to contain the answer, thereby improving the efficiency of the given NLU task (domain-dependent question answering). Additionally, Luo et al. [87] integrate KGs with LLMs to enable reasoning abilities for the task of knowledge-graph question answering (KGQA). They propose planning-retrieval-reasoning: LLMs first create a reasoning plan and then perform reasoning, which involves fetching reasoning paths from the KG using the generated plan. Hallucinations happen if the reasoning plan is incorrect. The aim is to distill the knowledge from KGs into LLMs to generate faithful relation paths as plans. Research studies such as [58, 90, 154] explore similar approaches, focusing on using LLMs for reasoning and question answering by incorporating knowledge graphs. However, when developing methods that combine KGs and LLMs, scalability must be a key consideration. Without effective scalability, integrating extensive knowledge bases or graphs can demand substantial resources, posing significant challenges for large-scale deployment [177].

Constrained-Decoding in LLMs: Recent advancements in natural language generation (NLG) have introduced constrained-decoding methods for LLMs [20, 44, 118], which apply ontological rules, among other constraints, to ensure that LLM outputs maintain logical consistency and conform to domain-specific structures [110]. It can also be seen as a way of integrating external knowledge with LLMs. The purpose of this integration is to provide a response adhering to the provided input syntax or semantics. Traditional constrained decoding with LLMs was applied at the syntax level; a typical example is the introduction of JSON mode [147]. However, the concept can also be extended to the level of semantics. A foundational work in this area is presented by Hokamp et al. [44], where they utilize lexicons to control the generation process of LLMs. This is an evolving area of research under the broader topic of synergies between graph-based knowledge sources and LLMs. Ontologies, the foundational framework for structuring and organizing knowledge graphs, formally represent the domain by serving as the Terminology Box (T-Box) that defines classes and their relationships [89]. These structured frameworks are instrumental in guiding the extraction and construction of domain-specific knowledge graphs and in constraining and refining LLM outputs to align with established schemas. Constrained-decoding leverages entities from the KG or the schema and

structure of ontologies to regulate LLM responses, ensuring relevance and adherence to domain-specific logic. Technically, constrained decoding encompasses methods for controlling the output tokens of LLMs using external knowledge sources, such as controlled vocabularies, taxonomies, structured ontologies, or domain-specific knowledge graphs. To summarize, ontologies and knowledge graphs provide the necessary structure and knowledge to anchor LLM-generated outputs, enabling the generation of domain-relevant language that aligns with the ontology's semantics.

4.2.3. Memory-based Systems

Retrieval-Augmented Generation (RAG): It is a kind of memory-oriented framework for LLMs, since it facilitates the external storage (*in the vector-database*) and dynamic retrieval of information. The work by Akbar et al. [3] discusses the use of vector databases and RAG frameworks to manage memory for conversational AI systems. The retrieval-augmented generation approach integrates dense vector search with LLM-based text generation, thereby improving response quality in knowledge-intensive applications. As its name implies, RAG retrieves contextually relevant information for a query using dense vector retrieval techniques. The retrieved information is then embedded into the LLM's prompt during text generation, improving both the factual accuracy and contextual relevance of the output. Rather than training or fine-tuning the LLM with vast amounts of knowledge, which consumes substantial resources and time, external knowledge can be dynamically retrieved using methods such as RAG. This reduces computational load and enables LLMs to scale efficiently while still benefiting from external knowledge [81]. It is most useful in scenarios where LLMs lag behind task-specific architectures. This approach enables the model to access and incorporate knowledge in real time, eliminating the need to store the entire knowledge base within its parameters. This significantly reduces computational overhead and enhances the model's decision provenance [77].

As highlighted above, RAG involves storing data in vector databases, such as Elasticsearch [33], ChromaDB [23], or others. Vector databases store vector representations of text, which are multidimensional arrays of numbers. They serve as the external memory. Several frameworks are available online that facilitate the seamless integration of RAG with LLMs. Examples include Hugging Face's RAG implementation, Haystack [28], and LangChain [93], among others. These frameworks provide tools and libraries to streamline the development of RAG pipelines, enabling efficient retrieval and context-enhanced text generation.

External knowledge integration into LLMs via RAG can be viewed as both parametric and non-parametric. Naive RAG implementations include using a pre-trained dense vector retriever, for example, instruct-embeddings [141] and connecting it with a pre-trained LLM. In this scenario, neither of the two models – the retrievers nor the LLMs – is further fine-tuned, i.e., following the non-parametric approach. The naive RAG approach focuses solely on the inference stage. In contrast, advanced RAG implementations offer two methods. The first method fine-tunes the dense-vector retriever for the specific task or domain. This is a non-parametric approach because the LLM's parameters remain unchanged. The second method fine-tunes both the retriever and the LLM for the specific task or domain. This is a parametric approach, in which the LLM's parameters and weights are adjusted to suit the use case. Some progress has also been made in pre-training RAG systems [39].

Further research on RAG implementations includes the following: RAG task categorization [182], where queries are classified into levels based on the type of external data required: explicit fact queries, implicit fact queries, interpretable rationale queries, and hidden rationale queries. This is done to ensure that the correct and most appropriate data is fetched for the given task and provided to the prompt as context. Zhao et al. [182] propose three methods of ingesting data into LLMs, i.e., context (providing the retrieved context directly to the LLM), small model (adding a small model trained on the domain to help external data integration with LLMs), and fine-tuning (fine-tuning the LLM). They also highlight the challenges of deploying data-augmented LLMs and believe that there is no one-size-fits-all solution for it. Anjum et al. [10] propose a framework called HALO for mitigating hallucinations in the medical question-answering systems to enhance reliability and accuracy. They use the RAG technique to integrate domain-specific information with the LLMs. The results show an increase in LLM accuracies from 44% to 65% for Llama-3.1 and from 56% to 70% for ChatGPT. Additional work in a similar line of research includes Hwang

et al. [54], who introduce Reliability RAG, an extension of traditional RAG systems designed to handle multiple data sources. Their approach focuses on estimating the reliability of various sources within the database. Similarly, Khan et al. [63] developed a PDF-based, LLM-powered RAG system, showcasing its application in processing and retrieving information from PDF documents.

Traditional RAG techniques have several limitations that can negatively impact the quality of generated responses. In particular, the retrieval component often returns irrelevant or low-quality results, which directly affects the final output. Key issues in retrieval include the lack of pre-retrieval query processing, the absence of post-retrieval enhancements such as reranking, and text chunking that disregards semantic boundaries. The generation component also faces constraints, including the context window limitation of LLMs and the inherent performance differences between smaller and larger models. To mitigate retrieval-related shortcomings, optimized approaches such as Advanced RAG and Modular RAG have been proposed, with a primary focus on enhancing retrieval quality. For a detailed discussion of these methods, see [2]. To address generation-related issues, such as context-window limitations and model capacity constraints, prior work has proposed methods, including positional interpolation, to extend context windows [19] and architectural remedies for generation breakdowns in long-context settings [46].

RAG research is ongoing and rapidly evolving, with emerging approaches, such as GraphRAG[10] [113] and other optimization methods, being proposed. Therefore, this work can be further extended to incorporate additional state-of-the-art research in this direction.

Other memory-based methods: Knowledge graphs and ontologies, as discussed in Section 4.2.2, serve as a form of semantic memory for large language models [3, p. 12]. Unlike the plain text-based embeddings typically used in RAG systems, they organize information into structured graph representations. This graphical organization enables graph-based retrieval, which goes beyond simple semantic-similarity search. Such capabilities form the key motivation behind the development of GraphRAG [113]. In a similar manner, Rasmussen et al. [126] adopt temporal knowledge graphs, i.e., graph structures that evolve over time and retain historical information, as a mechanism for episodic memory *(facts as episodes, changing over time)* in the implementation of GraphRAG. Beyond retrieval-centric approaches, MemGPT [108] introduces a complementary memory paradigm by simulating operating-system-like virtual memory management. Whereas RAG primarily augments LLMs with external embedding-based or semantic knowledge retrieved from document corpora, MemGPT equips them with working memory by storing and recalling prior interactions from external storage, thereby sustaining dialogue continuity and overcoming fixed context window limitations. Instead of modifying the model's parameters, MemGPT externalizes memory management: the LLM maintains a limited "active context" (analogous to RAM or short-term memory) while offloading less immediately relevant information into external storage (analogous to disk or long-term memory). This external memory may include prior conversation history, task states, or knowledge chunks, all stored outside the model's internal weights. When needed, MemGPT dynamically recalls and reinserts this information back into the context window, effectively integrating relevant working memory. In this way, MemGPT functions as another non-parametric memory-based method for LLMs. Similarly, Li et al. [84] investigate memory integration methods motivated by operating systems for LLMs. Refer to the survey [181] for more literature related to memory-based systems for large language models.

4.2.4. Tool usage and Function calling

The core of agentic AI systems lies in their ability to use external tools or functions. Tool usage refers to the ability of LLM-based systems to interact with external resources or services to perform tasks beyond their internal capabilities or training data. Function calling is a specific type of tool usage in which an LLM invokes a predefined function or API. A simple example of a tool is a calculator. Since LLMs do not perform actual calculations but instead predict the next word based on previous text, they might correctly answer simple questions like 2×2 because they have likely seen it in their training data. However, for less common numbers such as 2.356×0.627171111, the model may produce an incorrect result unless it

[10]Refer to the following Github repository for detailed research related to GraphRAG: https://github.com/pengboci/GraphRAG-Survey

Table 4
Comparison between Parametric and Non-Parametric Methods.

Aspect	Parametric Methods	Non-Parametric Methods
Computational Requirements	Computationally expensive and time-consuming; requires high-end GPU clusters for fine-tuning.	Less computationally expensive; RAG reduces load by avoiding full parameter-based storage.
Knowledge Updates	Requires modification of model weights or layers, which carries the risk of catastrophic forgetting.	Knowledge is stored externally, allowing for dynamic updates via external sources.
Explainability	Low explainability; knowledge embedded in weights.	High transparency and provenance can be traced.
Knowledge Representation	Implicitly encoded in parameters, memorization may not ensure accurate recall.	Explicitly stored in some external system, such as knowledge bases, retrieval systems, or an external model.
Specific Challenges	Frequency shock, attention imbalance, and difficulty in reverse retrieval.	Retrieval quality affects generation; context window limits usable information.
Flexibility	Static knowledge; retraining required for updates.	Adaptable through external access; supports real-time updates.

is connected to a calculator. This illustrates the concept of tool usage or function calling. Similarly, other tools could include a calendar or any external API, such as a weather service, or even a simple standalone Python function/method. For instance, early versions of ChatGPT often provided incorrect answers to questions like "What is today's date?", a problem that tool integration can address. The paper by [131], titled *Toolformer*, discusses how LLMs can learn to use external tools autonomously. Using such tools or calling APIs enables LLMs to access external information that is not stored in the model's internal weights or parameters. Frameworks like LangGraph [72] and LangChain [73] serve as orchestration engines for integrating various tools and functions with LLMs. There are methods proposed in the literature for enhancing the tool-calling capabilities of LLMs, for instance, Li et al. [78] propose MeCo, an adaptive decision-making strategy for external tool use. Similarly, Abdelaziz et al. [1] introduce *granite-20B-functioncalling*[11], a specialized model trained through a multi-task learning framework covering seven core function-calling tasks, including nested function calling, function chaining, next-based functions, parallel functions, and others. In the context of knowledge-intensive tasks, Hertling et al. [42] integrate knowledge graphs with LLMs through function calling. For more research related to tool usage, consult the following papers [26, 83, 122].

4.2.5. Reinforcement Learning

Reinforcement Learning (RL) is a subfield of machine learning in which an agent improves its performance by interacting with an environment and receiving feedback in the form of rewards or penalties [145]. In the context of Large Language Models (LLMs), RL can be employed to optimize model behavior across different tasks. Here, the model acts as the agent, while the environment is more broadly defined as the task setup and feedback loop; within this loop, the reward function or reward model, i.e., another machine learning system trained to evaluate the outputs of the LLMs, serves as an approximation of the environment's feedback. Importantly, Reinforcement Learning from Human Feedback (RLHF) extends this paradigm by training the environment's reward signal using human feedback, thereby aligning the model's outputs more closely with human expectations. Instruction-following is one such task in which the reward model guides the agent toward producing responses aligned with human preferences. A prominent example is ChatGPT, which was tuned using RLHF. In this case, the reward model was trained on human-labeled data that distinguished between high-quality and low-quality responses, as described in [106]. Following the similar area of research, Yan et al. [167], propose a variant of RLHF called Reinforcement Learning from Knowledge Graph Feedback (RLKGF), replacing human feedback as they are time-consuming and costly to accumulate, with a knowledge graph. They assume that the human reasoning process could be

[11] https://huggingface.co/ibm-granite/granite-20b-functioncalling

substituted by the reasoning paths in a knowledge graph. However, they also note that the knowledge in the KG represents human thinking, thereby aligning the model with human preferences. Their results show that RLKGF outperforms Reinforcement Learning from AI Feedback (RLAIF) [12, 74], another variant of RLHF in which the human feedback component is replaced by AI-generated feedback. Any discussion of RL in the context of LLMs cannot overlook the InstructGPT paper [106], although classifying it *and its related variants* as parametric or non-parametric is not straightforward. The reward model is trained externally, making that component non-parametric, while the LLMs policy is updated internally, meaning its weights are adjusted based on signals from the reward model, which is parametric. Taken together, this constitutes a hybrid approach. However, there are other RL-based methods that are purely non-parametric. One such method is *RL for prompt optimization*. For instance, Deng et al. [29] introduce RLPrompt, a discrete prompt-optimization method that harnesses reinforcement learning to generate optimal prompts that are transferable across models. Extending the work of RLPrompt, Kwon et al. [69] introduce StablePrompt, an automatic prompt tuning method based on reinforcement learning. In this framework, an agent model generates candidate prompts, while an anchor model is incorporated to stabilize policy updates. The target LLM, evaluated against the dataset, supplies reward signals that reflect the quality of the generated prompts and drive the optimization process.

5. Future Research Directions

As discussed in the sections above, there are limitations with LLMs which require further research. Additionally, the wide range of applications of LLMs in natural language understanding tasks presents numerous opportunities for future investigation. The following section outlines potential research directions for enhancing natural language understanding by leveraging LLMs and incorporating external knowledge to create highly efficient systems or models. These research directions emerged as key areas for further study, based on challenges highlighted in the surveyed literature and LLM optimization papers.

Explainability in Knowledge-Augmented LLMs: As LLMs integrate more external knowledge, ensuring their explainability becomes increasingly essential. Future research should focus on developing techniques to understand how knowledge is incorporated into the model's decision-making process. Additionally, improving the trustworthiness of applications built with LLMs will require methods to recover accurate citations for LLM-generated answers and effectively identify potential biases in the information they rely on [127]. For instance, improving the transparency of how knowledge graphs are integrated with LLMs and clarifying their reasoning processes [21] can enhance user trust.

Optimizing LLM finetuning: Future research can also address the issue of Frequency Shock (cf. Section 4.1.2) [60]. Other areas of improvement related to fine-tuning include mitigation strategies for overcoming attention imbalance [40]. Attention imbalance, as the name suggests, is a phenomenon observed in LLMs in which the attention mechanism disproportionately focuses on certain tokens. This can lead to the suppression of important information, such as factual knowledge stored in the model's parameters, from being used in generating responses.

Scalability and Efficiency Improvements in Knowledge Integration: A promising direction is improving the scalability and efficiency of knowledge-augmented LLMs. It is a critical area of research given the increasing demands for LLMs. Numerous strategies have been proposed to optimize LLMs, focusing on reducing computational cost and improving inference speed without compromising the model performance. These techniques include model compression, architectural advancements, and data efficiency methods, among others. Each of these techniques has its advantages and disadvantages and therefore requires further investigation and exploration [52, 80].

Agentic AI and Knowledge Integration: Future research could explore the integration of Agentic AI with LLMs, representing a significant advancement in artificial intelligence by combining the strengths of symbolic paradigms (symbolic representation and logic) with those of connectionist paradigms (neural networks). This integration has the potential to enhance knowledge integration and decision-making capabilities, enabling the development of autonomous agents that can navigate complex environments, reason, and learn from experiences. The synergy between these paradigms is crucial for enhancing the adaptability and reasoning capabilities of AI systems [133, 165].

Interactive Knowledge Retrieval and Integration: Interactive NLP systems (iNLP), where knowledge is dynamically retrieved and integrated during inference, offer a promising avenue for improving LLM-based NLU systems. For example, developing dynamically evolving knowledge graphs [18]. These systems engage in back-and-forth dialogues with external knowledge sources, refining their understanding of the task and accessing up-to-date information [161]. Combining LLM, Semantic Web (KGs and ontologies), and iNLP can also be a promising future research area.

6. Discussion

This paper explores LLM-based natural language understanding, particularly for knowledge-intensive tasks such as information extraction and knowledge graph construction in domain-specific contexts. It surveys methods for integrating domain-specific knowledge into LLMs to enhance accuracy and address common limitations of their out-of-the-box usage. It provides an overview of LLM-based NLU systems, discussing the technical details of how NLU tasks benefit from the use of LLMs since the advent of transformers. As LLMs have limitations, such as hallucinations, limited contextual understanding in specialized domains, and incomplete/outdated data, these limitations directly affect the performance of NLU systems. As part of the survey, the challenges of LLMs are discussed, and approaches to mitigate them are investigated. We acknowledge that while LLMs have demonstrated exceptional performance across a wide range of NLU tasks, integrating external knowledge sources, such as ontologies and knowledge graphs, among others, is a critical pathway to address issues of factual accuracy, domain-specific reasoning, and ambiguity resolution. The discussion emphasizes the importance of integrating external knowledge to enhance factual accuracy, minimize hallucinations, and optimize performance on domain-specific tasks. Two primary approaches for incorporating knowledge into LLMs were explored, namely parametric and non-parametric methods. Parametric methods encode knowledge within a model's parameters by updating its weights through techniques such as pre-training, fine-tuning, steering, knowledge editing, embedding-based approaches, and knowledge distillation. These approaches offer advantages such as higher accuracy in knowledge extraction and manipulation. However, they are computationally intensive and prone to challenges such as catastrophic forgetting, knowledge conflicts, and limited explainability. In contrast, non-parametric approaches, such as prompting strategies, knowledge-based techniques, memory-augmented systems (e.g., RAG), and tool-usage or function-calling methods, operate without extensive model training. These methods offer greater explainability, improved hardware efficiency, and increased flexibility, making them well-suited for scenarios where adaptability and explainability are crucial. Table 4 provides a comparison of these approaches across other relevant dimensions.

The choice of integration technique ultimately depends on the specific use case and the nature and volume of the data. Furthermore, the method's efficiency can only be determined through experimentation and thorough evaluation, as with other AI methods and algorithms. As research into improving knowledge-intensive NLU systems with LLMs is ongoing, there is ample room for experimentation and further investigation into developing autonomous methods to enhance the explainability of LLM-generated outputs and to explore the integration of Agentic AI with LLMs. Combining fields such as the Semantic Web, NLP, and interactive NLP (iNLP) to create systems that incorporate real-time human feedback, thereby adding a human-in-the-loop dimension, represents a significant opportunity to advance LLM-based NLU systems.

7. Conclusion

The survey investigates techniques of integrating external knowledge with LLMs. The taxonomy and the Python script [12] used for searching and crawling the seed papers can be found at [166]. The study seeks to provide insights to enhance the effectiveness of the out-of-the-box LLMs in handling domain-specific and knowledge-intensive tasks. Its emphasis lies in examining research on LLM-driven natural language understanding systems, while highlighting the inherent challenges of LLMs that constrain the performance of such NLU applications. Approaches for overcoming the common limitations of LLMs, including hallucinations, lack of contextual understanding, and outdated data, are discussed in detail. These approaches suggest integrating external knowledge to achieve better factual accuracy with LLMs for knowledge-intensive tasks, such as information extraction and knowledge graph construction. The investigation highlights that while parametric methods for integrating knowledge into LLMs have shown a positive impact on performance, they face challenges, including a lack of explainability and the extensive use of hardware resources during model training. Non-parametric approaches, which rely on external storage systems for knowledge retention, offer improved explainability and allow the provenance of results to be traced. Moreover, implementing non-parametric approaches is less computationally expensive. However, the decision of which approach to pick depends on the particular problem. Trade-offs must be made between performance and hardware requirements. Lastly, future research directions for improving LLM-based NLU systems are discussed.

References

[1] I. Abdelaziz, K. Basu, M. Agarwal, S. Kumaravel, M. Stallone, R. Panda, Y. Rizk, G.P.S. Bhargav, M. Crouse, C. Gunasekara, S. Ikbal, S. Joshi, H. Karanam, V. Kumar, A. Munawar, S. Neelam, D. Raghu, U. Sharma, A.M. Soria, D. Sreedhar, P. Venkateswaran, M. Unuvar, D.D. Cox, S. Roukos, L.A. Lastras and P. Kapanipathi, Granite-Function Calling Model: Introducing Function Calling Abilities via Multi-task Learning of Granular Tasks, in: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, F. Dernoncourt, D. Preoţiuc-Pietro and A. Shimorina, eds, Association for Computational Linguistics, Miami, Florida, US, 2024, pp. 1131–1139. doi:10.18653/v1/2024.emnlp-industry.85.

[2] M. Abo El-Enen, S. Saad and T. Nazmy, A survey on retrieval-augmentation generation (RAG) models for healthcare applications, *Neural Computing and Applications* **37**(33) (2025), 28191–28267. doi:10.1007/s00521-025-11666-9.

[3] N.A. Akbar, R. Dembani, B. Lenzitti and D. Tegolo, RAG-Driven Memory Architectures in Conversational LLMs—A Literature Review With Insights Into Emerging Agriculture Data Sharing, *IEEE Access* **13** (2025), 123855–123880. doi:10.1109/access.2025.3589241.

[4] M. Alam, G.A. Gesese and P.-H. Paris, Neurosymbolic Methods for Dynamic Knowledge Graphs (2024). doi:10.48550/arxiv.2409.04572.

[5] L. Albtosh, Challenges and Limitations of Using LLMs in Software Security, *Advances in information security, privacy, and ethics book series* (2024), 439–464. ISBN 9798369393130. doi:10.4018/979-8-3693-9311-6.ch012.

[6] Z. Allen-Zhu and Y. Li, Physics of Language Models: Part 3.1, Knowledge Storage and Extraction (2023). doi:10.48550/ARXIV.2309.14316.

[7] Z. Allen-Zhu and Y. Li, Physics of Language Models: Part 3.1, Knowledge Storage and Extraction (2023). doi:10.48550/arXiv.2309.14316.

[8] Z. Allen-Zhu and Y. Li, Physics of Language Models: Part 3.2, Knowledge Manipulation (2023). doi:10.48550/ARXIV.2309.14402.

[9] M.K. Anjali and A.P. Babu, Ambiguities in natural language processing, *International Journal of Innovative Research in Computer and Communication Engineering* **2**(5) (2014), 392–394.

[10] S. Anjum, H. Zhang, W. Zhou, E.J. Paek, X. Zhao and Y. Feng, HALO: Hallucination Analysis and Learning Optimization to Empower LLMs with Retrieval-Augmented Context for Guided Clinical Decision Making (2024). doi:10.48550/arxiv.2409.10011.

[11] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak and Z. Ives, *DBpedia: A Nucleus for a Web of Open Data*, in: *The Semantic Web*, Springer Berlin Heidelberg, 2007, pp. 722–735, Springer. ISSN 1611-3349. ISBN 9783540762980. doi:10.1007/978-3-540-76298-0_52.

[12] The script can be reused in the future for extending this work with the latest developments.

[12] Y. Bai, S. Kadavath, S. Kundu, A. Askell, J. Kernion, A. Jones, A. Chen, A. Goldie, A. Mirhoseini, C. McKinnon, C. Chen, C. Olsson, C. Olah, D. Hernandez, D. Drain, D. Ganguli, D. Li, E. Tran-Johnson, E. Perez, J. Kerr, J. Mueller, J. Ladish, J. Landau, K. Ndousse, K. Lukosuite, L. Lovitt, M. Sellitto, N. Elhage, N. Schiefer, N. Mercado, N. DasSarma, R. Lasenby, R. Larson, S. Ringer, S. Johnston, S. Kravec, S.E. Showk, S. Fort, T. Lanham, T. Telleen-Lawton, T. Conerly, T. Henighan, T. Hume, S.R. Bowman, Z. Hatfield-Dodds, B. Mann, D. Amodei, N. Joseph, S. McCandlish, T. Brown and J. Kaplan, Constitutional AI: Harmlessness from AI Feedback, *arXiv preprint arXiv:2212.08073* (2022). doi:10.48550/ARXIV.2212.08073.

[13] E. Ben Zaken, Y. Goldberg and S. Ravfogel, BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models, in: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Association for Computational Linguistics, 2022, pp. 1–9. doi:10.18653/v1/2022.acl-short.1.

[14] Z. Bouhoun, A. Allali, R. Cocci, M. Assaad, A. Plancon, F. Godest, K. Kondratenko, J. Rodriguez, F. Vitillo, O. Malhomme, L.B. Bechet and R. Plana, CurieLM: Enhancing Large Language Models for Nuclear Domain Applications, *EPJ Web of Conferences* **302** (2024), 17006–17006. doi:10.1051/epjconf/202430217006.

[15] T. Brown, B. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell et al., Language models are few-shot learners, *Advances in neural information processing systems* **33** (2020), 1877–1901.

[16] T. Brown, B. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever and D. Amodei, Language Models are Few-Shot Learners, in: *Advances in Neural Information Processing Systems*, Vol. 33, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan and H. Lin, eds, Curran Associates, Inc., 2020, pp. 1877–1901. https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[17] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras and I. Androutsopoulos, LEGAL-BERT: The Muppets straight out of Law School, in: *Findings of the Association for Computational Linguistics: EMNLP 2020*, T. Cohn, Y. He and Y. Liu, eds, Association for Computational Linguistics, Online, 2020, pp. 2898–2904. doi:10.18653/v1/2020.findings-emnlp.261.

[18] R. Chen, W. Jiang, C. Qin, I.S. Rawal, C. Tan, D. Choi, B. Xiong and B. Ai, LLM-Based Multi-Hop Question Answering with Knowledge Graph Integration in Evolving Environments, in: *Findings of the Association for Computational Linguistics: EMNLP 2024*, Y. Al-Onaizan, M. Bansal and Y.-N. Chen, eds, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 14438–14451. doi:10.18653/v1/2024.findings-emnlp.844.

[19] S. Chen, S. Wong, L. Chen and Y. Tian, Extending Context Window of Large Language Models via Positional Interpolation, *arXiv preprint arXiv:2306.15595* (2023). doi:10.48550/ARXIV.2306.15595.

[20] X. Chen, Z. Yang and X. Wan, Relation-constrained decoding for text generation, *Advances in Neural Information Processing Systems* **35** (2022), 26804–26819.

[21] Z. Chen, J. Chen, Y. Chen, H. Yu, A.K. Singh and M. Sra, LMExplainer: Grounding Knowledge and Explaining Language Models (2023). doi:10.48550/arXiv.2303.16537.

[22] N. Choudhary and C.K. Reddy, Complex Logical Reasoning over Knowledge Graphs using Large Language Models (2023). doi:10.48550/ARXIV.2305.01157.

[23] Chroma, Chroma: Open Source Embedding Database, 2025, Accessed: 2025-01-09. https://github.com/chroma-core/chroma.

[24] M. Cleti and P. Jano, Hallucinations in LLMs: Types, Causes, and Approaches for Enhanced Reliability, Oct, 2024. doi:10.31219/osf.io/tj93u.

[25] R. Cohen, M. Geva, J. Berant and A. Globerson, Crawling The Internal Knowledge-Base of Language Models (2023), 1856–1869. doi:10.18653/v1/2023.findings-eacl.139.

[26] S. Cui, A. He, S. Xu, H. Zhang, Y. Wang, Q. Zhang, Y. Wang and B. Xu, Self-Guided Function Calling in Large Language Models via Stepwise Experience Recall, in: *Findings of the Association for Computational Linguistics: EMNLP 2025*, C. Christodoulopoulos, T. Chakraborty, C. Rose and V. Peng, eds, Association for Computational Linguistics, Suzhou, China, 2025, pp. 10842–10854. ISBN 979-8-89176-335-7. doi:10.18653/v1/2025.findings-emnlp.574.

[27] L.Y. da Costa and J.B.d. Oliveira e Souza Filho, Adapting LLMs to New Domains: A Comparative Study of Fine-Tuning and RAG strategies for Portuguese QA Tasks, in: *Anais do XV Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana (STIL 2024)*, STIL 2024, Sociedade Brasileira de Computação, 2024, pp. 267–277. doi:10.5753/stil.2024.245443.

[28] deepset, Haystack: An End-to-End Framework for Building NLP Applications, 2025, Accessed: 2025-01-09. https://haystack.deepset.ai/.

[29] M. Deng, J. Wang, C.-P. Hsieh, Y. Wang, H. Guo, T. Shu, M. Song, E. Xing and Z. Hu, RLPrompt: Optimizing Discrete Text Prompts with Reinforcement Learning, in: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Y. Goldberg, Z. Kozareva and Y. Zhang, eds, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022, pp. 3369–3391. doi:10.18653/v1/2022.emnlp-main.222.

[30] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: *Proceedings of the 2019 Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), J. Burstein, C. Doran and T. Solorio, eds, Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. doi:10.18653/v1/n19-1423.

[31] G. Dong, R. Li, S. Wang, Y. Zhang, Y. Xian and W. Xu, Bridging the KB-Text Gap: Leveraging Structured Knowledge-aware Pre-training for KBQA, in: *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, Association for Computing Machinery, New York, NY, USA, 2023, pp. 3854–3859–. ISBN 9798400701245. doi:10.1145/3583780.3615150.

[32] Y. Dong, R. Mu, Y. Zhang, S. Sun, T. Zhang, C. Wu, G. Jin, Y. Qi, J. Hu, J. Meng, S. Bensalem and X. Huang, Safeguarding large language models: a survey, *Artificial Intelligence Review* **58**(12) (2025), 382. doi:10.1007/s10462-025-11389-2.

[33] Elastic, Elastic: Search, Observe, Protect, 2025, Accessed: 2025-01-09. https://www.elastic.co/.

[34] W. Fan, Y. Ding, L. Ning, S. Wang, H. Li, D. Yin, T.-S. Chua and Q. Li, A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models, in: *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, Association for Computing Machinery, New York, NY, USA, 2024, pp. 6491–6501–. ISBN 9798400704901. doi:10.1145/3637528.3671470.

[35] C. Feng, X. Zhang and Z. Fei, Knowledge Solver: Teaching LLMs to Search for Domain Knowledge from Knowledge Graphs (2023). doi:10.48550/ARXIV.2309.03118.

[36] D. Fensel, U. Şimşek, K. Angele, E. Huaman, E. Kärle, O. Panasiuk, I. Toma, J. Umbrich, A. Wahler, D. Fensel et al., Introduction: what is a knowledge graph?, *Knowledge graphs: Methodology, tools and selected use cases* (2020), 1–10.

[37] J. Fields, K. Chovanec and P. Madiraju, A Survey of Text Classification With Transformers: How Wide? How Large? How Long? How Accurate? How Expensive? How Safe?, *IEEE Access* **12** (2024), 6518–6531. doi:10.1109/ACCESS.2024.3349952.

[38] D. Gao, K. Chen, B. Chen, H. Dai, L. Jin, W. Jiang, W. Ning, S. Yu, Q. Xuan, X. Cai, L. Yang and Z. Wang, LLMs-based machine translation for E-commerce, *Expert Systems with Applications* **258** (2024), 125087. doi:10.1016/j.eswa.2024.125087.

[39] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang and H. Wang, Retrieval-augmented generation for large language models: A survey (2023). doi:10.48550/ARXIV.2312.10997.

[40] G.R. Ghosal, T. Hashimoto and A. Raghunathan, Understanding Finetuning for Factual Knowledge Extraction, Cornell University, 2024. doi:10.48550/arxiv.2406.14785.

[41] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey and N.A. Smith, Don't Stop Pretraining: Adapt Language Models to Domains and Tasks, in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter and J. Tetreault, eds, Association for Computational Linguistics, Online, 2020, pp. 8342–8360. doi:10.18653/v1/2020.acl-main.740.

[42] S. Hertling and H. Sack, Towards Large Language Models Interacting with Knowledge Graphs Via Function Calling (2024).

[43] G. Hinton, O. Vinyals and J. Dean, Distilling the Knowledge in a Neural Network, *arXiv preprint arXiv:1503.02531* (2015). doi:10.48550/ARXIV.1503.02531.

[44] C. Hokamp and Q. Liu, Lexically Constrained Decoding for Sequence Generation Using Grid Beam Search, in: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, 2017. doi:10.18653/v1/p17-1141.

[45] H.Y. Hongkang Yang, Z.L. Zehao Lin, W.W. Wenjin Wang, H.W. Hao Wu, Z.L. Zhiyu Li, B.T. Bo Tang, W.W. Wenqiang Wei, J.W. Jinbo Wang, Z.T. Zeyun Tang, S.S. Shichao Song, C.X. Chenyang Xi, Y.Y. Yu Yu, K.C. Kai Chen, F.X. Feiyu Xiong, L.T. Linpeng Tang and W.E. Weinan E, Memory^3: Language Modeling with Explicit Memory, *Journal of Machine Learning* **3**(3) (2024), 300–346. doi:10.4208/jml.240708.

[46] P. Hosseini, I. Castro, I. Ghinassi and M. Purver, Efficient Solutions For An Intriguing Failure of LLMs: Long Context Window Does Not Mean LLMs Can Analyze Long Sequences Flawlessly, in: *Proceedings of the 31st International Conference on Computational Linguistics*, O. Rambow, L. Wanner, M. Apidianaki, H. Al-Khalifa, B.D. Eugenio and S. Schockaert, eds, Association for Computational Linguistics, Abu Dhabi, UAE, 2025, pp. 1880–1891. https://aclanthology.org/2025.coling-main.128/.

[47] J. Howard and S. Ruder, Universal Language Model Fine-tuning for Text Classification, in: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, I. Gurevych and Y. Miyao, eds, Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 328–339. doi:10.18653/v1/P18-1031.

[48] E.J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang and W. Chen, LoRA: Low-Rank Adaptation of Large Language Models (2021). doi:10.48550/arXiv.2106.09685.

[49] L. Hu, Z. Liu, Z. Zhao, L. Hou, L. Nie and J. Li, A Survey of Knowledge Enhanced Pre-Trained Language Models, *IEEE Trans. on Knowl. and Data Eng.* **36**(4) (2023), 1413–1430–. doi:10.1109/TKDE.2023.3310002.

[50] L. Hu, Z. Liu, Z. Zhao, L. Hou, L. Nie and J. Li, A Survey of Knowledge Enhanced Pre-Trained Language Models, *IEEE Transactions on Knowledge and Data Engineering* **36**(4) (2024), 1413–1430. doi:10.1109/TKDE.2023.3310002.

[51] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin and T. Liu, A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions, *ACM Trans. Inf. Syst.* (2024), Just Accepted. doi:10.1145/3703155.

[52] S. Huang, K. Yang, S. Qi and R. Wang, When large language model meets optimization, *Swarm and Evolutionary Computation* **90** (2024), 101663. doi:10.1016/j.swevo.2024.101663.
[53] Y. Huang, K. Tang and M. Chen, Leveraging Large Language Models for Enhanced NLP Task Performance through Knowledge Distillation and Optimized Training Strategies (2024). doi:10.48550/ARXIV.2402.09282.
[54] J. Hwang, J. Park, H. Park, S. Park and J. Ok, Retrieval-Augmented Generation with Estimation of Source Reliability (2024). doi:10.48550/arxiv.2410.22954.
[55] G. Ilharco, M.T. Ribeiro, M. Wortsman, S. Gururangan, L. Schmidt, H. Hajishirzi and A. Farhadi, Editing Models with Task Arithmetic (2022). doi:10.48550/ARXIV.2212.04089.
[56] R. Ishigaki, J. Suzuki, M. Shuzo and E. Maeda, Knowledge Editing of Large Language Models Unconstrained by Word Order, in: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, X. Fu and E. Fleisig, eds, Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 159–169. ISBN 979-8-89176-097-4. doi:10.18653/v1/2024.acl-srw.23.
[57] P. Jain and M. Lapata, Integrating Large Language Models with Graph-based Reasoning for Conversational Question Answering (2024). doi:10.48550/ARXIV.2407.09506.
[58] Y. Ji, K. Wu, J. Li, W. Chen, M. Zhong, X. Jia and M. Zhang, Retrieval and Reasoning on KGs: Integrate Knowledge Graphs into Large Language Models for Complex Question Answering, in: *Findings of the Association for Computational Linguistics: EMNLP 2024*, Association for Computational Linguistics, 2024, pp. 7598–7610. doi:10.18653/v1/2024.findings-emnlp.446.
[59] D. Jin, E. Pan, N. Oufattole, W.-H. Weng, H. Fang and P. Szolovits, What Disease Does This Patient Have? A Large-Scale Open Domain Question Answering Dataset from Medical Exams, *Applied Sciences* **11**(14) (2021), 6421. doi:10.3390/app11146421.
[60] M. Kazemi, S. Mittal and D. Ramachandran, Understanding Finetuning for Factual Knowledge Extraction from Language Models (2023). doi:10.48550/ARXIV.2301.11293.
[61] A. Keluskar, A. Bhattacharjee and H. Liu, Do LLMs Understand Ambiguity in Text? A Case Study in Open-world Question Answering, in: *2024 IEEE International Conference on Big Data (BigData)*, IEEE, 2024, pp. 7485–7490. doi:10.1109/bigdata62323.2024.10825265.
[62] T. Kerner, Domain-Specific Pretraining of Language Models: A Comparative Study in the Medical Field (2024). doi:10.48550/ARXIV.2407.14076.
[63] A. Khan, M.T. Hasan, K.K. Kemell, J. Rasku and P. Abrahamsson, Developing Retrieval Augmented Generation (RAG) based LLM Systems from PDFs: An Experience Report (2024). doi:10.48550/arxiv.2410.15944.
[64] L. Khanbutayeva, The problem of structural ambiguity in Psycholinguistics, *Revista Tempos e Espaços em Educação* **13**(32) (2020), 1–17. doi:10.20952/REVTEE.V13I32.14675.
[65] J.O. Krugmann and J. Hartmann, Sentiment Analysis in the Age of Generative AI, *Customer Needs and Solutions* **11**(1) (2024). doi:10.1007/s40547-024-00143-4.
[66] S. Kukreja, T. Kumar, A. Purohit, A. Dasgupta and D. Guha, A Literature Survey on Open Source Large Language Models, in: *Proceedings of the 2024 7th International Conference on Computers in Management and Business*, ICCMB 2024, ACM, 2024, pp. 133–143. doi:10.1145/3647782.3647803.
[67] C.S. Kulkarni, The Evolution of Large Language Models in Natural Language Understanding, *Journal of Artificial Intelligence, Machine Learning and Data Science* **1**(4) (2023), 49–53. doi:10.51219/jaimld/chinmay-shripad-kulkarni/28.
[68] P. Kumar, Large language models (LLMs): survey, technical frameworks, and future challenges, *Artificial Intelligence Review* **57**(10) (2024), 260. doi:10.1007/s10462-024-10888-y.
[69] M. Kwon, G. Kim, J. Kim, H. Lee and J. Kim, StablePrompt: Automatic Prompt Tuning using Reinforcement Learning for Large Language Model, in: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Y. Al-Onaizan, M. Bansal and Y.-N. Chen, eds, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 9868–9884. doi:10.18653/v1/2024.emnlp-main.551.
[70] W. Lai, V. Hangya and A. Fraser, Style-Specific Neurons for Steering LLMs in Text Style Transfer, in: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Y. Al-Onaizan, M. Bansal and Y.-N. Chen, eds, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 13427–13443. doi:10.18653/v1/2024.emnlp-main.745.
[71] T.A. Lamb, A. Davies, A. Paren, P.H.S. Torr and F. Pinto, Focus On This, Not That! Steering LLMs With Adaptive Feature Specification (2024). doi:10.48550/ARXIV.2410.22944.
[72] LangChain-AI, langgraph: Build resilient language agents as graphs, n.d., Accessed: 2025-11-15.
[73] LangChain-AI, LangChain: Build context-aware reasoning applications, n.d., Accessed: 2025-11-15.
[74] H. Lee, S. Phatale, H. Mansoor, T. Mesnard, J. Ferret, K. Lu, C. Bishop, E. Hall, V. Carbune, A. Rastogi and S. Prakash, RLAIF vs. RLHF: Scaling Reinforcement Learning from Human Feedback with AI Feedback, *arXiv preprint arXiv:2309.00267* (2023). doi:10.48550/ARXIV.2309.00267.
[75] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C.H. So and J. Kang, BioBERT: a pre-trained biomedical language representation model for biomedical text mining, *Bioinformatics* **36**(4) (2020), 1234–1240.

[76] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov and L. Zettlemoyer, BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension, in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter and J. Tetreault, eds, Association for Computational Linguistics, Online, 2020, pp. 7871–7880. doi:10.18653/v1/2020.acl-main.703.

[77] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel et al., Retrieval-augmented generation for knowledge-intensive nlp tasks, *Advances in Neural Information Processing Systems* **33** (2020), 9459–9474.

[78] W. Li, D. Li, K. Dong, C. Zhang, H. Zhang, W. Liu, Y. Wang, R. Tang and Y. Liu, Adaptive Tool Use in Large Language Models with Meta-Cognition Trigger, in: *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, Vienna, Austria, 2025, pp. 13346–13370. doi:10.18653/v1/2025.acl-long.655.

[79] X.L. Li and P. Liang, Prefix-Tuning: Optimizing Continuous Prompts for Generation, in: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Association for Computational Linguistics, 2021, pp. 4582–4597. doi:10.18653/v1/2021.acl-long.353.

[80] X. Li, Y. Ma, Y. Huang, X. Wang, Y. Lin and C. Zhang, Synergized Data Efficiency and Compression (SEC) Optimization for Large Language Models (2025). doi:10.20944/preprints202409.0662.v3.

[81] X. Li, S. Mei, Z. Liu, Y. Yan, S. Wang, Y. Shi, Z. Zeng, H. Chen, G. Yu, Z. Liu, M. Sun and C. Xiong, RAG-DDR: Optimizing Retrieval-Augmented Generation Using Differentiable Data Rewards (2024). doi:10.48550/arxiv.2410.13509.

[82] Y. Li, H. Zhao, H. Jiang, Y. Pan, Z. Liu, Z. Wu, P. Shu, J. Tian, T. Yang, S. Xu, Y. Lyu, P. Blenk, J. Pence, J. Rupram, E. Banu, N. Liu, S. Wang, W. Song, X. Zhai, K. Song, D. Zhu, B.S. Li, X. Wang and T. Liu, Large Language Models for Manufacturing (2024). doi:10.48550/arxiv.2410.21418.

[83] Z. Li, Z. Chen, M. Ross, P. Huber, S. Moon, Z. Lin, X. Dong, A. Sagar, X. Yan and P. Crook, Large Language Models as Zero-shot Dialogue State Tracker through Function Calling, in: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, L.-W. Ku, A. Martins and V. Srikumar, eds, Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 8688–8704. doi:10.18653/v1/2024.acl-long.471.

[84] Z. Li, S. Song, C. Xi, H. Wang, C. Tang, S. Niu, D. Chen, J. Yang, C. Li, Q. Yu, J. Zhao, Y. Wang, P. Liu, Z. Lin, P. Wang, J. Huo, T. Chen, K. Chen, K. Li, Z. Tao, H. Lai, H. Wu, B. Tang, Z. Wang, Z. Fan, N. Zhang, L. Zhang, J. Yan, M. Yang, T. Xu, W. Xu, H. Chen, H. Wang, H. Yang, W. Zhang, Z.-Q.J. Xu, S. Chen and F. Xiong, MemOS: A Memory OS for AI System, *arXiv preprint arXiv:2507.03724* (2025). doi:10.48550/ARXIV.2507.03724.

[85] J. Liu, Q. Mao, W. Jiang and J. Li, KnowFormer: Revisiting Transformers for Knowledge Graph Reasoning (2024). doi:10.48550/ARXIV.2409.12865.

[86] Y. Liu, Y. Yao, J.-F. Ton, X. Zhang, R. Guo, H. Cheng, Y. Klochkov, M.F. Taufiq and H. Li, Trustworthy LLMs: a Survey and Guideline for Evaluating Large Language Models' Alignment (2023). doi:10.48550/ARXIV.2308.05374.

[87] L. Luo, Y.-F. Li, G. Haffari and S. Pan, Reasoning on Graphs: Faithful and Interpretable Large Language Model Reasoning (2023). doi:10.48550/ARXIV.2310.01061.

[88] Z. Luo, Y. Yang, R. Qi, Z. Fang, Y. Guo and Y. Wang, Incorporating Large Language Models into Named Entity Recognition: Opportunities and Challenges, in: *2023 4th International Conference on Computer, Big Data and Artificial Intelligence (ICCBD+AI)*, 2023, pp. 429–433. doi:10.1109/ICCBD-AI62252.2023.00079.

[89] Z. Luo, Y. Wang, W. Ke, R. Qi, Y. Guo and P. Wang, Boosting LLMS with Ontology-Aware Prompt for Ner Data Augmentation, in: *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 12361–12365. doi:10.1109/ICASSP48485.2024.10446860.

[90] J. Ma, Z. Gao, Q. Chai, W. Sun, P. Wang, H. Pei, J. Tao, L. Song, J. Liu, C. Zhang et al., Debate on Graph: a Flexible and Reliable Reasoning Framework for Large Language Models (2024). doi:10.48550/ARXIV.2409.03155.

[91] S. Maes, Fixing Reference Hallucinations of LLMs (2025). doi:10.31219/osf.io/u38w4_v2.

[92] S. Mangrulkar, S. Gugger, L. Debut, Y. Belkada, S. Paul and B. Bossan, PEFT: State-of-the-art Parameter-Efficient Fine-Tuning methods, 2022.

[93] V. Mavroudis, LangChain (2024). doi:10.20944/preprints202411.0566.v1.

[94] T.R. McIntosh, T. Susnjak, T. Liu, P. Watters and M.N. Halgamuge, The Inadequacy of Reinforcement Learning From Human Feedback—Radicalizing Large Language Models via Semantic Vulnerabilities, *IEEE Transactions on Cognitive and Developmental Systems* **16**(4) (2024), 1561–1574. doi:10.1109/tcds.2024.3377445.

[95] M. McTear, S. Varghese Marokkie and Y. Bi, *A Comparative Study of Chatbot Response Generation: Traditional Approaches Versus Large Language Models*, in: *Knowledge Science, Engineering and Management*, Springer Nature Switzerland, 2023, pp. 70–79. ISSN 1611-3349. ISBN 9783031402869. doi:10.1007/978-3-031-40286-9_7.

[96] K. Meng, D. Bau, A. Andonian and Y. Belinkov, Locating and editing factual associations in gpt, *Advances in neural information processing systems* **35** (2022), 17359–17372.

[97] K. Meng, A.S. Sharma, A. Andonian, Y. Belinkov and D. Bau, Mass-Editing Memory in a Transformer, *arXiv preprint arXiv:2210.07229* (2022). doi:10.48550/ARXIV.2210.07229.

[98] T. Mihaylov, P. Clark, T. Khot and A. Sabharwal, Can a suit of armor conduct electricity? a new dataset for open book question answering, in: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2018. doi:10.18653/v1/d18-1260.

[99] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain and J. Gao, Large language models: A survey, *arXiv preprint arXiv:2402.06196* (2024).

[100] E. Mitchell, C. Lin, A. Bosselut, C. Finn and C.D. Manning, Fast Model Editing at Scale, *arXiv preprint arXiv:2110.11309* (2021). doi:10.48550/ARXIV.2110.11309.

[101] A. Mukanova, M. Milosz, A. Dauletkaliyeva, A. Nazyrova, G. Yelibayeva, D. Kuzin and L. Kussepova, LLM-Powered Natural Language Text Processing for Ontology Enrichment, *Applied Sciences* **14**(13) (2024), 5860. doi:10.3390/app14135860.

[102] B. Muralidharan, H. Beadles, R. Marzban and K.S. Mupparaju, Knowledge AI: Fine-tuning NLP Models for Facilitating Scientific Knowledge Extraction and Understanding (2024). doi:10.48550/arxiv.2408.04651.

[103] A. Nagar, V. Schlegel, T.-T. Nguyen, H. Li, Y. Wu, K. Binici and S. Winkler, LLMs are not Zero-Shot Reasoners for Biomedical Information Extraction (2024). doi:10.48550/ARXIV.2408.12249.

[104] N.I. of Standards and T. (NIST), Reuters Corpora (RCV1, RCV2, TRC2), 2004, Accessed: 2025-11-06.

[105] J. Omeliyanenko, A. Zehe, A. Hotho and D. Schlör, CapsKG: Enabling Continual Knowledge Integration in Language Models for Automatic Knowledge Graph Completion, in: *The Semantic Web – ISWC 2023: 22nd International Semantic Web Conference, Athens, Greece, November 6–10, 2023, Proceedings, Part I*, Springer-Verlag, Berlin, Heidelberg, 2023, pp. 618–636–. ISBN 978-3-031-47239-8. doi:10.1007/978-3-031-47240-4_33.

[106] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C.L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike and R. Lowe, Training language models to follow instructions with human feedback, *Advances in neural information processing systems* **35** (2022), 27730–27744. doi:10.48550/ARXIV.2203.02155.

[107] O. Ovadia, M. Brief, M. Mishaeli and O. Elisha, Fine-Tuning or Retrieval? Comparing Knowledge Injection in LLMs, in: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Y. Al-Onaizan, M. Bansal and Y.-N. Chen, eds, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 237–250. doi:10.18653/v1/2024.emnlp-main.15.

[108] C. Packer, S. Wooders, K. Lin, V. Fang, S.G. Patil, I. Stoica and J.E. Gonzalez, MemGPT: Towards LLMs as Operating Systems (2023). doi:10.48550/ARXIV.2310.08560.

[109] P.H. Paiola, G.L. Garcia, J.R.R. Manesco, M. Roder, D. Rodrigues and J.P. Papa, Adapting LLMs for the Medical Domain in Portuguese: A Study on Fine-Tuning and Model Evaluation (2024). doi:10.48550/ARXIV.2410.00163.

[110] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang and X. Wu, Unifying Large Language Models and Knowledge Graphs: A Roadmap, *IEEE Transactions on Knowledge and Data Engineering* **36**(7) (2024), 3580–3599. doi:10.1109/TKDE.2024.3352100.

[111] N. Pantha, M. Ramasubramanian, I. Gurung, M. Maskey and R. Ramachandran, Challenges in Guardrailing Large Language Models for Science (2024). doi:10.48550/arxiv.2411.08181.

[112] B. Peng, E. Chersoni, Y.-Y. Hsu and C.-R. Huang, Is Domain Adaptation Worth Your Investment? Comparing BERT and FinBERT on Financial Tasks, in: *Proceedings of the Third Workshop on Economics and Natural Language Processing*, U. Hahn, V. Hoste and A. Stent, eds, Association for Computational Linguistics, Punta Cana, Dominican Republic, 2021, pp. 37–44. doi:10.18653/v1/2021.econlp-1.5.

[113] B. Peng, Y. Zhu, Y. Liu, X. Bo, H. Shi, C. Hong, Y. Zhang and S. Tang, Graph Retrieval-Augmented Generation: A Survey (2024). doi:10.48550/ARXIV.2408.08921.

[114] A. Petrov, P.H. Torr and A. Bibi, When Do Prompting and Prefix-Tuning Work? A Theory of Capabilities and Limitations, *arXiv preprint arXiv:2310.19698* (2023). doi:10.48550/ARXIV.2310.19698.

[115] D.K. Pham and Q.B. Vo, Towards Reliable Medical Question Answering: Techniques and Challenges in Mitigating Hallucinations in Language Models (2024). doi:10.48550/arxiv.2408.13808.

[116] Y. Pinter and M. Elhadad, Emptying the Ocean with a Spoon: Should We Edit Models?, in: *Findings of the Association for Computational Linguistics: EMNLP 2023*, H. Bouamor, J. Pino and K. Bali, eds, Association for Computational Linguistics, Singapore, 2023, pp. 15164–15172. doi:10.18653/v1/2023.findings-emnlp.1012.

[117] A. Piñeiro-Martín, F.-J. Santos-Criado, C. García-Mateo, L. Docío-Fernández and M.d.C. López-Pérez, Context Is King: Large Language Models' Interpretability in Divergent Knowledge Scenarios, *Applied Sciences* **15**(3) (2025), 1192. doi:10.3390/app15031192.

[118] M. Post and D. Vilar, Fast lexically constrained decoding with dynamic beam allocation for neural machine translation (2018). doi:10.48550/arXiv.1804.06609.

[119] C. Poth, H. Sterz, I. Paul, S. Purkayastha, L. Engländer, T. Imhof, I. Vulić, S. Ruder, I. Gurevych and J. Pfeiffer, Adapters: A Unified Library for Parameter-Efficient and Modular Transfer Learning, in: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Y. Feng and E. Lefever, eds, Association for Computational Linguistics, Singapore, 2023, pp. 149–160. doi:10.18653/v1/2023.emnlp-demo.13.

[120] Z. Qiang, K. Taylor, W. Wang and J. Jiang, OAEI-LLM: A Benchmark Dataset for Understanding Large Language Model Hallucinations in Ontology Matching (2024). doi:10.48550/arxiv.2409.14038.

[121] S. Qiao, Y. Ou, N. Zhang, X. Chen, Y. Yao, S. Deng, C. Tan, F. Huang and H. Chen, Reasoning with Language Model Prompting: A Survey, in: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber and N. Okazaki, eds, Association for Computational Linguistics, Toronto, Canada, 2023, pp. 5368–5393. doi:10.18653/v1/2023.acl-long.294.

[122] S. Qin, Y. Zhu, L. Mu, S. Zhang and X. Zhang, Meta-Tool: Unleash Open-World Function Calling Capabilities of General-Purpose Large Language Models, in: *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, W. Che, J. Nabende, E. Shutova and M.T. Pilehvar, eds, Association for Computational Linguistics, Vienna, Austria, 2025, pp. 30653–30677. ISBN 979-8-89176-251-0. doi:10.18653/v1/2025.acl-long.1481.

[123] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever et al., Improving language understanding by generative pre-training (2018).

[124] H. Rajabzadeh, M. Valipour, T. Zhu, M. Tahaei, H.J. Kwon, A. Ghodsi, B. Chen and M. Rezagholizadeh, QDyLoRA: Quantized Dynamic Low-Rank Adaptation for Efficient Large Language Model Tuning (2024). doi:10.48550/ARXIV.2402.10462.

[125] A. Rajasekharan, Y. Zeng, P. Padalkar and G. Gupta, Reliable Natural Language Understanding with Large Language Models and Answer Set Programming, in: *International Conference on Logic Programming*, Vol. 385, Open Publishing Association, 2023, pp. 274–287. ISSN 2075-2180. doi:10.4204/eptcs.385.27.

[126] P. Rasmussen, P. Paliychuk, T. Beauvais, J. Ryan and D. Chalef, Zep: A Temporal Knowledge Graph Architecture for Agent Memory, *arXiv preprint arXiv:2501.13956* (2025). doi:10.48550/ARXIV.2501.13956.

[127] J. Rorseth, P. Godfrey, L. Golab, D. Srivastava and J. Szlichta, Towards Explainability in Retrieval-Augmented LLMs, in: *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, 2024, pp. 5669–5670. doi:10.1109/ICDE60146.2024.00466.

[128] V. Sanh, L. Debut, J. Chaumond and T. Wolf, DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, *arXiv preprint arXiv:1910.01108* (2019). doi:10.48550/ARXIV.1910.01108.

[129] E. Sanu, T.K. Amudaa, P. Bhat, G. Dinesh, A.U. Kumar Chate and R.K. P, Limitations of Large Language Models, in: *2024 8th International Conference on Computational System and Information Technology for Sustainable Solutions (CSITSS)*, 2024, pp. 1–6. doi:10.1109/CSITSS64042.2024.10817070.

[130] Y. Saxena, S. Chopra and A. Tripathi, Evaluating Consistency and Reasoning Capabilities of Large Language Models, Cornell University, 2024. doi:10.48550/arxiv.2404.16478.

[131] T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, M. Lomeli, E. Hambro, L. Zettlemoyer, N. Cancedda and T. Scialom, Toolformer: Language models can teach themselves to use tools, *Advances in Neural Information Processing Systems* **36** (2023), 68539–68551.

[132] A. Sedova, R. Litschko, D. Frassinelli, B. Roth and B. Plank, To Know or Not To Know? Analyzing Self-Consistency of Large Language Models under Ambiguity (2024). doi:10.48550/arxiv.2407.17125.

[133] A. Sharma, Bridging Paradigms: The Integration of Symbolic and Connectionist AI in LLM-Driven Autonomous Agents, *Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023* **6**(1) (2024), 138–150.

[134] R.K. Sharma and M. Joshi, An analytical study and review of open source chatbot framework, rasa, *Int. J. Eng. Res* **9**(06) (2020), 1011–1014. doi:10.17577/ijertv9is060723.

[135] L. Shi, Z. Tang, N. Zhang, X. Zhang and Z. Yang, A Survey on Employing Large Language Models for Text-to-SQL Tasks (2024). doi:10.48550/ARXIV.2407.15186.

[136] C. Shimizu and P. Hitzler, Accelerating knowledge graph and ontology engineering with large language models, *Journal of Web Semantics* **85** (2025), 100862. doi:10.1016/j.websem.2025.100862.

[137] O.P. Singh and D.M.E. Patil, Analysis of Ambiguity, Vagueness, Fuzziness, Uncertainty, Possibility and Probability in the Natural Language Semantics with Fuzzy Logic, *International Research Journal on Advanced Engineering Hub (IRJAEH)* **2**(05) (2024), 1478–1483. doi:10.47392/irjaeh.2024.0204.

[138] S. Singh, N. Zaidi and A. Singh, Deep learning for natural language understanding: A review of recent advances, *International journal of applied research* (2018). doi:10.22271/allresearch.2018.v4.i10d.11459.

[139] L. Some, W. Yang, M. Bain and B.H. Kang, A Comprehensive Survey on Integrating Large Language Models with Knowledge-Based Methods (2025). doi:10.2139/ssrn.5111497.

[140] H. Soudani, E. Kanoulas and F. Hasibi, Fine Tuning vs. Retrieval Augmented Generation for Less Popular Knowledge, in: *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, SIGIR-AP 2024, Association for Computing Machinery, New York, NY, USA, 2024, pp. 12–22–. ISBN 9798400707247. doi:10.1145/3673791.3698415.

[141] H. Su, W. Shi, J. Kasai, Y. Wang, Y. Hu, M. Ostendorf, W.-t. Yih, N.A. Smith, L. Zettlemoyer and T. Yu, One Embedder, Any Task: Instruction-Finetuned Text Embeddings, in: *Findings of the Association for Computational Linguistics: ACL 2023*, A. Rogers, J. Boyd-Graber and N. Okazaki, eds, Association for Computational Linguistics, Toronto, Canada, 2023, pp. 1102–1121. doi:10.18653/v1/2023.findings-acl.71.

[142] N. Sulaiman and F. Hamzah, Evaluation of Transfer Learning and Adaptability in Large Language Models with the GLUE Benchmark, *Authorea Preprints* (2024). doi:10.36227/techrxiv.171077989.99407624/v1.

[143] S. Sun, Y. Liu, S. Wang, D. Iter, C. Zhu and M. Iyyer, PEARL: Prompting Large Language Models to Plan and Execute Actions Over Long Documents, in: *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, Y. Graham and M. Purver, eds, Association for Computational Linguistics, St. Julian's, Malta, 2024, pp. 469–486. doi:10.18653/v1/2024.eacl-long.29.

[144] T. Susnjak, P. Hwang, N. Reyes, A.L. Barczak, T. McIntosh and S. Ranathunga, Automating Research Synthesis with Domain-Specific Large Language Model Fine-Tuning, *ACM Transactions on Knowledge Discovery from Data* **19**(3) (2025), 1–39. doi:10.1145/3715964.

[145] R.S. Sutton and A.G. Barto, Reinforcement Learning: An Introduction, *IEEE Transactions on Neural Networks* **9**(5) (1998), 1054–1054. doi:10.1109/tnn.1998.712192.

[146] A. Talmor, O. Yoran, R.L. Bras, C. Bhagavatula, Y. Goldberg, Y. Choi and J. Berant, CommonsenseQA 2.0: Exposing the Limits of AI through Gamification (2022). doi:10.48550/ARXIV.2201.05320.

[147] Z.R. Tam, C.-K. Wu, Y.-L. Tsai, C.-Y. Lin, H.-y. Lee and Y.-N. Chen, Let Me Speak Freely? A Study On The Impact Of Format Restrictions On Large Language Model Performance., in: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, F. Dernoncourt, D. Preoţiuc-Pietro and A. Shimorina, eds, Association for Computational Linguistics, Miami, Florida, US, 2024, pp. 1218–1236. doi:10.18653/v1/2024.emnlp-industry.91.

[148] S. Tian, Y. Luo, T. Xu, C. Yuan, H. Jiang, C. Wei and X. Wang, KG-Adapter: Enabling Knowledge Graph Integration in Large Language Models through Parameter-Efficient Fine-Tuning, in: *Findings of the Association for Computational Linguistics: ACL 2024*, L.-W. Ku, A. Martins and V. Srikumar, eds, Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 3813–3828. doi:10.18653/v1/2024.findings-acl.229.

[149] C. Vasantharajan, K.Z. Tun, H. Thi-Nga, S. Jain, T. Rong and C.E. Siong, MedBERT: A Pre-trained Language Model for Biomedical Named Entity Recognition, in: *2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, IEEE, 2022, pp. 1482–1488. doi:10.23919/apsipaasc55919.2022.9980157.

[150] A. Vaswani, Attention is all you need, *Advances in Neural Information Processing Systems* (2017).

[151] D. Vrandečić and M. Krötzsch, Wikidata: a free collaborative knowledgebase, *Commun. ACM* **57**(10) (2014), 78–85–. doi:10.1145/2629489.

[152] F. Wang, R. Bao, S. Wang, W. Yu, Y. Liu, W. Cheng and H. Chen, InfuserKI: Enhancing Large Language Models with Knowledge Graphs via Infuser-Guided Knowledge Integration, in: *Findings of the Association for Computational Linguistics: EMNLP 2024*, Y. Al-Onaizan, M. Bansal and Y.-N. Chen, eds, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 3675–3688. doi:10.18653/v1/2024.findings-emnlp.209.

[153] H. Wang and Y.-F. Li, Large Language Model Empowered by Domain-Specific Knowledge Base for Industrial Equipment Operation and Maintenance, in: *2023 5th International Conference on System Reliability and Safety Engineering (SRSE)*, 2023, pp. 474–479. doi:10.1109/SRSE59585.2023.10336112.

[154] K. Wang, F. Duan, S. Wang, P. Li, Y. Xian, C. Yin, W. Rong and Z. Xiong, Knowledge-Driven CoT: Exploring Faithful Reasoning in LLMs for Knowledge-intensive Question Answering (2023). doi:10.48550/ARXIV.2308.13259.

[155] M. Wang, Y. Yao, Z. Xu, S. Qiao, S. Deng, P. Wang, X. Chen, J.-C. Gu, Y. Jiang, P. Xie, F. Huang, H. Chen and N. Zhang, Knowledge Mechanisms in Large Language Models: A Survey and Perspective, in: *Findings of the Association for Computational Linguistics: EMNLP 2024*, Y. Al-Onaizan, M. Bansal and Y.-N. Chen, eds, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 7097–7135. doi:10.18653/v1/2024.findings-emnlp.416.

[156] P. Wang, N. Zhang, B. Tian, Z. Xi, Y. Yao, Z. Xu, M. Wang, S. Mao, X. Wang, S. Cheng, K. Liu, Y. Ni, G. Zheng and H. Chen, EasyEdit: An Easy-to-use Knowledge Editing Framework for Large Language Models, in: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Y. Cao, Y. Feng and D. Xiong, eds, Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 82–93. doi:10.18653/v1/2024.acl-demos.9.

[157] R. Wang, D. Tang, N. Duan, Z. Wei, X. Huang, J. Ji, G. Cao, D. Jiang and M. Zhou, K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters, in: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, C. Zong, F. Xia, W. Li and R. Navigli, eds, Association for Computational Linguistics, Online, 2021, pp. 1405–1418. doi:10.18653/v1/2021.findings-acl.121.

[158] S. Wang, Y. Zhu, H. Liu, Z. Zheng, C. Chen and J. Li, Knowledge Editing for Large Language Models: A Survey, *ACM Computing Surveys* **57**(3) (2024), 1–37. doi:10.1145/3698590.

[159] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang and M. Zhou, Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers, *Advances in neural information processing systems* **33** (2020), 5776–5788.

[160] Y. Wang, B. Jiang, Y. Luo, D. He, P. Cheng and L. Gao, Reasoning on Efficient Knowledge Paths:Knowledge Graph Guides Large Language Model for Domain Question Answering (2024). doi:10.48550/ARXIV.2404.10384.

[161] Z. Wang, G. Zhang, K. Yang, N. Shi, W. Zhou, S. Hao, G. Xiong, Y. Li, M.Y. Sim, X. Chen, Q. Zhu, Z. Yang, A. Nik, Q. Liu, C. Lin, S. Wang, R. Liu, W. Chen, K. Xu, D. Liu, Y. Guo and J. Fu, Interactive Natural Language Processing (2023). doi:10.48550/ARXIV.2305.13246.

[162] Z. Wang, Z. Chu, T.V. Doan, S. Ni, M. Yang and W. Zhang, History, development, and principles of large language models: an introductory survey, *AI and Ethics* **5**(3) (2025), 1955–1971.

[163] Z. Wang, H. Wang, B.P. Danek, Y. Li, C. Mack, H. Poon, S. Wang, P. Rajpurkar and J. Sun, A Perspective for Adapting Generalist AI to Specialized Medical AI Applications and Their Challenges (2024). doi:10.48550/arxiv.2411.00024.

[164] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E.H. Chi, Q.V. Le and D. Zhou, Chain-of-thought prompting elicits reasoning in large language models, in: *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Curran Associates Inc., Red Hook, NY, USA, 2022. ISBN 9781713871088.

[165] H. Xiong, Z. Wang, X. Li, J. Bian, Z. Xie, S. Mumtaz, A. Al-Dulaimi and L.E. Barnes, Converging Paradigms: The Synergy of Symbolic and Connectionist AI in LLM-Empowered Autonomous Agents (2024). doi:10.48550/ARXIV.2407.08516.

[166] I. Yadav, Script and Taxonomy for External Knowledge Integration in Large Language Models: A Survey on Methods, Challenges, and Future Directions, Zenodo, 2025. doi:10.5281/zenodo.15064852.

[167] L. Yan, C. Tang, Y. Guan, H. Wang, S. Wang, H. Liu, Y. Yang and J. Jiang, RLKGF: Reinforcement Learning from Knowledge Graph Feedback Without Human Annotations, in: *Findings of the Association for Computational Linguistics: ACL 2025*, W. Che, J. Nabende, E. Shutova and M.T. Pilehvar, eds, Association for Computational Linguistics, Vienna, Austria, 2025, pp. 6619–6633. ISBN 979-8-89176-256-5. doi:10.18653/v1/2025.findings-acl.344.

[168] R. Yan, L. Sun, F. Wang and X. Zhang, K-XLNet: A General Method for Combining Explicit Knowledge with Language Model Pretraining (2021). doi:10.48550/ARXIV.2104.10649.

[169] X. Yan, Y. Xiao and Y. Jin, Generative Large Language Models Explained [AI-eXplained], *IEEE Computational Intelligence Magazine* **19**(4) (2024), 45–46. doi:10.1109/mci.2024.3431454.

[170] S. Yang, F. Chen, Y. Yang and Z. Zhu, A Study on Semantic Understanding of Large Language Models from the Perspective of Ambiguity Resolution, in: *Proceedings of the 2023 International Joint Conference on Robotics and Artificial Intelligence*, JCRAI 2023, ACM, 2023, pp. 165–170. doi:10.1145/3632971.3632973.

[171] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov and Q.V. Le, *XLNet: generalized autoregressive pretraining for language understanding*, in: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Curran Associates Inc., Red Hook, NY, USA, 2019.

[172] L. Yao, C. Mao and Y. Luo, KG-BERT: BERT for Knowledge Graph Completion (2019). doi:10.48550/ARXIV.1909.03193.

[173] J. Ye, X. Chen, N. Xu, C. Zu, Z. Shao, S. Liu, Y. Cui, Z. Zhou, C. Gong, Y. Shen, J. Zhou, S. Chen, T. Gui, Q. Zhang and X. Huang, A Comprehensive Capability Analysis of GPT-3 and GPT-3.5 Series Models (2023). doi:10.48550/ARXIV.2303.10420.

[174] W. Ye, Q. Zhang, X. Zhou, W. Hu, C. Tian and J. Cheng, Correcting Factual Errors in LLMs via Inference Paths Based on Knowledge Graph, in: *2024 International Conference on Computational Linguistics and Natural Language Processing (CLNLP)*, IEEE, 2024, pp. 12–16. doi:10.1109/clnlp64123.2024.00011.

[175] F. Zait and N. Zarour, Addressing Lexical and Semantic Ambiguity in Natural Language Requirements, in: *2018 Fifth International Symposium on Innovation in Information and Communication Technology (ISIICT)*, 2018, pp. 1–7. doi:10.1109/ISIICT.2018.8613726.

[176] C. Zhang, B. Peng, X. Sun, Q. Niu, J. Liu, K. Chen, M. Li, P. Feng, Z. Bi, M. Liu, Y. Zhang, F. Cheng, C.H. Yin, L. Yan and T. Wang, From Word Vectors to Multimodal Embeddings: Techniques, Applications, and Future Directions For Large Language Models (2024). doi:10.48550/arxiv.2411.05036.

[177] N. Zhang, Y. Yao, B. Tian, P. Wang, S. Deng, M. Wang, Z. Xi, S. Mao, J. Zhang, Y. Ni, S. Cheng, Z. Xu, X. Xu, J.-C. Gu, Y. Jiang, P. Xie, F. Huang, L. Liang, Z. Zhang, X. Zhu, J. Zhou and H. Chen, A Comprehensive Study of Knowledge Editing for Large Language Models (2024). doi:10.48550/ARXIV.2401.01286.

[178] Q. Zhang, C. Singh, L. Liu, X. Liu, B. Yu, J. Gao and T. Zhao, Tell Your Model Where to Attend: Post-hoc Attention Steering for LLMs (2023). doi:10.48550/ARXIV.2311.02262.

[179] T. Zhang, C. Wang, N. Hu, M. Qiu, C. Tang, X. He and J. Huang, DKPLM: Decomposable Knowledge-Enhanced Pre-trained Language Model for Natural Language Understanding, *Proceedings of the AAAI Conference on Artificial Intelligence* **36**(10) (2022), 11703–11711. doi:10.1609/aaai.v36i10.21425.

[180] T. Zhang, R. Xu, C. Wang, Z. Duan, C. Chen, M. Qiu, D. Cheng, X. He and W. Qian, Learning Knowledge-Enhanced Contextual Language Representations for Domain Natural Language Understanding, in: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino and K. Bali, eds, Association for Computational Linguistics, Singapore, 2023, pp. 15663–15676. doi:10.18653/v1/2023.emnlp-main.969.

[181] Z. Zhang, Q. Dai, X. Bo, C. Ma, R. Li, X. Chen, J. Zhu, Z. Dong and J.-R. Wen, A Survey on the Memory Mechanism of Large Language Model-based Agents, *ACM Transactions on Information Systems* **43**(6) (2025), 1–47. doi:10.1145/3748302.

[182] S. Zhao, Y. Yang, Z. Wang, Z. He, L.K. Qiu and L. Qiu, Retrieval Augmented Generation (RAG) and Beyond: A Comprehensive Survey on How to Make your LLMs use External Data More Wisely (2024). doi:10.48550/ARXIV.2409.14924.

[183] Y. Zhao, L. Yan, W. Sun, G. Xing, S. Wang, C. Meng, Z. Cheng, Z. Ren and D. Yin, Improving the Robustness of Large Language Models via Consistency Alignment, in: *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti and N. Xue, eds, ELRA and ICCL, Torino, Italia, 2024, pp. 8931–8941.