Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.061263

REVIEW

A Critical Review of Methods and Challenges in Large Language Models

Milad Moradi[1,*], Ke Yan[2], David Colwell[2], Matthias Samwald[3] and Rhona Asgari[1]

[1]AI Research, Tricentis, Vienna, 1220, Austria
[2]AI Research, Tricentis, Sydney, NSW 2010, Australia
[3]Institute of Artificial Intelligence, Center for Medical Statistics, Informatics, and Intelligent Systems, Medical University of Vienna, Vienna, 1090, Austria

*Corresponding Author: Milad Moradi. Email: m.moradi-vastegani@tricentis.com

Received: 20 November 2024; Accepted: 09 January 2025; Published: 17 February 2025

ABSTRACT: This critical review provides an in-depth analysis of Large Language Models (LLMs), encompassing their foundational principles, diverse applications, and advanced training methodologies. We critically examine the evolution from Recurrent Neural Networks (RNNs) to Transformer models, highlighting the significant advancements and innovations in LLM architectures. The review explores state-of-the-art techniques such as in-context learning and various fine-tuning approaches, with an emphasis on optimizing parameter efficiency. We also discuss methods for aligning LLMs with human preferences, including reinforcement learning frameworks and human feedback mechanisms. The emerging technique of retrieval-augmented generation, which integrates external knowledge into LLMs, is also evaluated. Additionally, we address the ethical considerations of deploying LLMs, stressing the importance of responsible and mindful application. By identifying current gaps and suggesting future research directions, this review provides a comprehensive and critical overview of the present state and potential advancements in LLMs. This work serves as an insightful guide for researchers and practitioners in artificial intelligence, offering a unified perspective on the strengths, limitations, and future prospects of LLMs.

KEYWORDS: Large language models; artificial intelligence; natural language processing; machine learning; generative artificial intelligence

1 Introduction

Generative Artificial Intelligence (AI) has rapidly advanced, transforming AI through models like the Generative Pre-trained Transformer (GPT) series [1,2]. With large neural networks, novel Machine Learning (ML) algorithms, and extensive training datasets, these models excel in understanding and generating human-like text. Their accessibility and open-source frameworks have democratized generative Large Language Models (LLMs), enabling their integration across sectors such as chatbots, healthcare, and finance [3–6]. This review provides a comprehensive analysis of LLMs, examining their foundational principles, applications, methodologies, and challenges. By evaluating existing methods, identifying research gaps, and suggesting future directions, it aims to offer coherent insights valuable to researchers and practitioners.

In the early 2010s, Recurrent Neural Networks (RNNs) demonstrated effectiveness in sequential processing for capturing contextual dependencies and generating coherent text [7]. However, they struggled with long-range dependencies, vanishing or exploding gradients, and slow processing [8]. Transformers revolutionized text generation by introducing attention mechanisms that capture context across entire

sequences simultaneously [9]. Models like GPT outperformed RNNs with parallelization, improved long-term dependency handling, and enhanced linguistic modeling through multi-headed self-attention [10]. The capabilities of Large Language Models (LLMs) have grown exponentially due to advancements in transformer architectures, massive text datasets, and computational power [11,12]. These developments, along with increased parameter counts, enable LLMs to excel in complex NLP tasks. Widely adopted across fields like healthcare, finance, education, and technology, LLMs demonstrate versatility and transformative impact [3,4,13,14]. Table 1 highlights prominent LLMs developed since transformers' advent.

Table 1: Well-known LLMs developed and released since 2018, along with the number of parameters each one has

Year	Short name	Full name	Parameters
2018	GPT-1	Generative pre-trained transformer 1	117 million
	BERT-large	Bidirectional encoder representation from transformers	340 million
2019	XLNet-large	–	340 million
	GPT-2	Generative pre-trained transformer 2	1.5 billion
2020	T5	Text-to-text transfer transformer	11 billion
	GPT-3	Generative pre-trained transformer 3	175 billion
2021	LaMDA	Language model for dialogue applications	137 billion
2022	PaLM-1	Pathways language model 1	540 billion
	BLOOM	BigScience large open-science open-access multilingual language model	176 billion
2023	LLaMA	Large language model meta AI	65 billion
	Claude-1	–	93 billion
	Claude-2	–	340 billion
	PaLM-2	Pathways language model 2	137 billion
	GPT-4	Generative pre-trained transformer 4	>1 trillion
2024	Gemini 1	–	1.5 trillion
	Mistral	–	7 billion
	Gemini 1.5	–	2.4 trillion

LLMs are widely applied across domains, excelling in NLP tasks like text generation, translation, summarization, and sentiment analysis. They power chatbots and virtual assistants in conversational systems, support medical diagnosis and patient interaction in healthcare, and enhance finance through automated trading, fraud detection, and customer support. In education, they enable personalized learning and tutoring, while in technology and creative arts, they aid in code generation, content creation, music composition, and visual art generation.

LLMs have demonstrated significant practical value in solving real-world problems across various domains. For instance, in healthcare, LLMs like GPT-4 are used to draft patient discharge summaries, reducing administrative burdens on medical professionals while ensuring accuracy in medical documentation [3,15]. In finance, models such as BloombergGPT assist analysts by generating detailed sentiment analyses of market trends based on news and financial reports, enabling more informed investment decisions [14]. In the field of education, tools powered by LLMs like ChatGPT provide personalized tutoring, helping students understand complex topics through interactive question-and-answer sessions. Furthermore, in software development, LLM-based systems like Copilot aid programmers by offering real-time

code suggestions and debugging assistance, thereby accelerating development processes. These examples underscore the transformative potential of LLMs in automating routine tasks, enhancing decision-making processes, and fostering innovation across diverse sectors.

Table 2 provides a taxonomy of these applications. LLMs significantly improve efficiency and productivity by automating tasks, enhancing accessibility via translation and summarization, and fostering innovation in creative industries. They reduce business costs and support informed decision-making in healthcare and finance. Educational outcomes are improved through interactive tutoring. However, challenges like bias, privacy, security, transparency, and environmental impact must be addressed to ensure responsible deployment.

Table 2: Taxonomy and classification of applications of LLMs

Domain	LLM applications
Education and research	• Tutoring systems providing personalized learning experiences. • Summarization of academic papers and generation of research hypotheses.
Healthcare and medical	• Medical documentation automation. • Analysis and generation of patient information leaflets.
Finance and economics	• Sentiment analysis of financial reports and news. • Automated financial advising and report generation.
Technology and software development	• Code generation and assistance in software development. • Bug detection and automated code documentation.
Legal and compliance	• Automated contract review and legal document analysis. • Compliance monitoring through the analysis of communications and documents.
Marketing and advertising	• Generation of personalized marketing content. • Social media content creation and management.
Entertainment and gaming	• Creating dynamic dialogues for non-player characters in video games. • Scriptwriting assistance for movies and TV shows.
Human resources	• Resume screening and job matching. • Automated generation of job descriptions.
Public relations and communications	• Crisis management through sentiment analysis of social media. • Automated press release generation.
Customer service	• Chatbots for handling customer inquiries. • Automated email response generation.
Content creation and journalism	• Automated generation of news articles and reports. • Writing assistance for creative writing, scripts, and advertising copy.
Translation and linguistics	• Real-time translation services. • Dialect and language preservation through linguistic analysis.

This review explores the lifecycle of LLM-powered applications, covering model architecture selection, pre-training, domain adaptation, alignment with human preferences, and application integration. It examines state-of-the-art methodologies and best practices in designing, developing, and deploying LLMs. Key challenges, including sensitivity analysis, uncertainty quantification, and error improvement, are highlighted. The review aims to provide a comprehensive understanding of LLMs and identify opportunities for future research and innovation.

2 Pre-Training LLMs

Pre-training large language models (LLMs) involves training on extensive text data to learn patterns, contextual relationships, and language structures [16]. This process develops a generalized understanding of language, stored in the model's parameters, which act as its memory. Larger parameter counts enhance the model's memory and ability to handle complex tasks [17]. Parameters are optimized during pre-training to minimize loss and improve accuracy. Once pre-trained, LLMs can be fine-tuned on task-specific datasets, leveraging their broad linguistic knowledge for diverse applications.

2.1 Model Architectures and Pre-Training Objectives

LLMs are typically pre-trained in a self-supervised manner, where no labeled training samples are utilized to direct the training process [18]. The choice of pretraining objectives significantly influences the performance and capabilities of LLMs, and these objectives vary depending on the model architecture and intended tasks [19]. A transformer language model can be composed of an encoder, a decoder, or both components, each serving distinct purposes and having specific advantages and limitations.

2.1.1 Encoder-Decoder Models

Encoder-decoder models, or sequence-to-sequence (seq2seq) models, use an encoder to process input sequences and a decoder to generate outputs, excelling in tasks like translation, question answering, and summarization. Their pretraining combines masked language modeling with seq2seq reconstruction, enabling contextual representation learning and autoregressive generation. Notable examples include T5 [20] and BART [21]. Despite their effectiveness, encoder-decoder models face scalability challenges when expanded to billions of parameters, prompting a shift toward encoder-only and decoder-only models in modern LLM design. Addressing these scalability issues while maintaining performance on complex tasks remains a critical research focus. Fig. 1 illustrates the seq2seq architecture.

2.1.2 Encoder-Only Models

Encoder-only models, or autoencoders, use self-attention to compress input sequences into dense contextual representations and reconstruct the input [22]. Pre-trained with a masked language modeling objective, they predict masked tokens based on context. Examples include BERT [10] and RoBERTa [23], excelling in tasks like text classification, sentiment analysis, and Named Entity Recognition (NER). While effective for understanding and representation tasks, encoder-only models have limited generative capabilities compared to decoder-only models.

2.1.3 Decoder-Only Models

Decoder-only models, or autoregressive models, generate outputs by attending to previously generated tokens and conditioning on the context [24]. Pre-trained with the causal language modeling objective, they predict the next token based on preceding tokens, ensuring unidirectional causality. These models excel in

text generation tasks, with examples including GPT [25], Chinchilla [26], BLOOM [27], and LLaMA [28]. Despite their dominance in generative tasks, decoder-only models require vast data and computational resources and often struggle with maintaining coherence over long sequences or avoiding repetitive outputs.

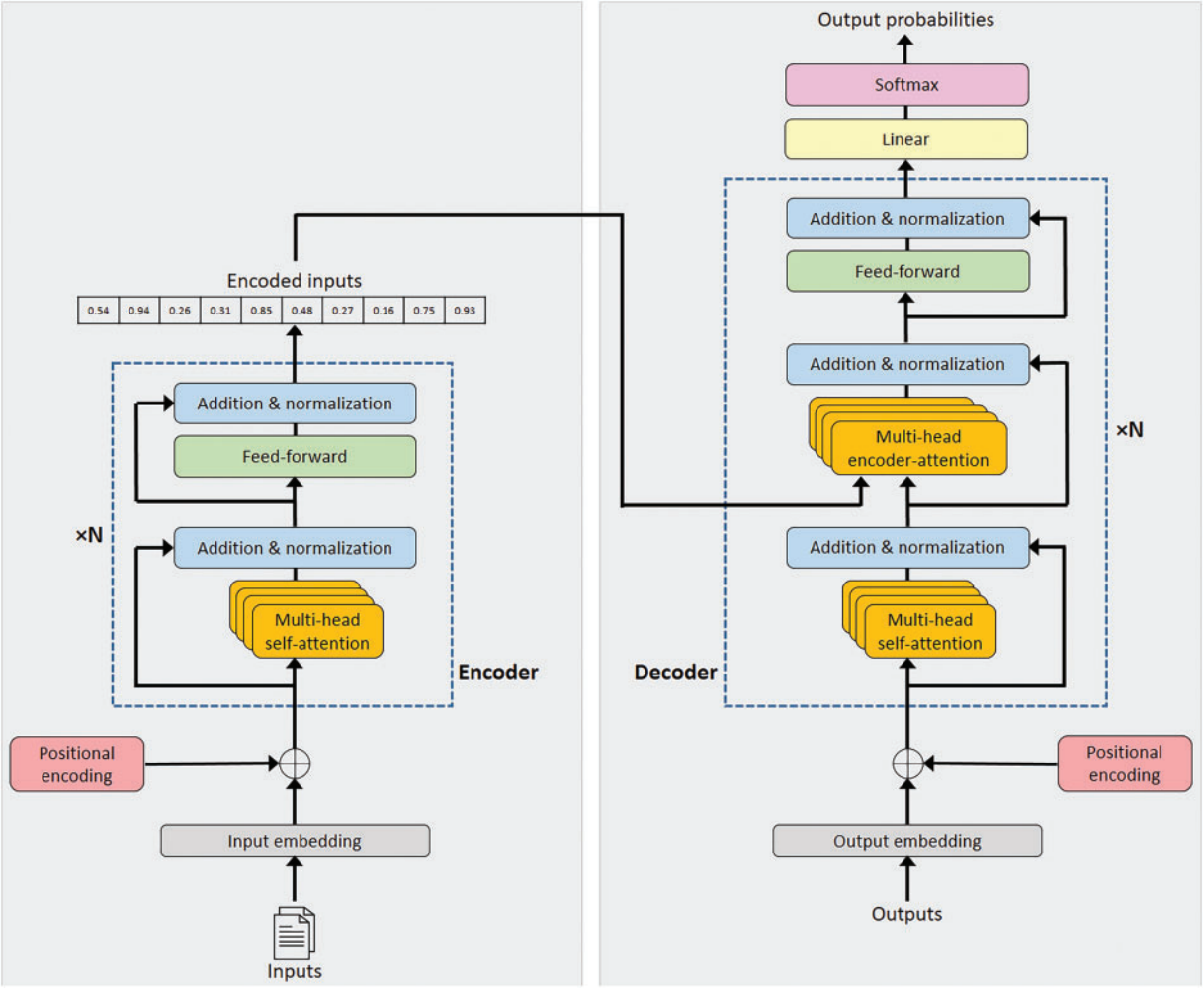

Figure 1: The overall architecture of an encoder-decoder transformer language model. The encoder and decoder components consist of several encoder and decoder blocks. In the encoder component, the input is first mapped to embeddings, which are numerical vectors. The embeddings are combined with positional encodings, then multi-head self-attention computes a representation conditioning on other words in the sequence. Other computations such as addition, normalization, and feed-forward layers perform subsequent computations resulting in the final encoded input. The decoder component receives the outputs generated in the previous time steps, converts them to embeddings, combines them with positional encoding, and passes them through self-attention, encoder-attention, addition, normalization and feed-forward layers. Linear transformations and the softmax function are finally applied to have probabilities over the vocabulary for the next output token. Encoder-only and decoder-only models are comprised of multiple encoder or decoder blocks, respectively

Comparative evaluation of these architectures reveals distinct advantages and limitations. Encoder-decoder models excel in seq2seq tasks but face scalability challenges. Encoder-only models are efficient in understanding tasks but are limited in generative capabilities. Decoder-only models are unparalleled in text generation but require extensive computational resources and data. A critical gap in current research

is the integration of strengths from each model architecture to develop more versatile and efficient LLMs. Additionally, there is a need for innovative pretraining objectives that can further enhance the performance and scalability of these models. Addressing these gaps will be crucial for the advancement of LLMs and their application across diverse domains.

3 Domain Adaptation

Domain adaptation in LLMs refers to the process of adjusting these models to perform effectively in specific domains of interest or on particular tasks. LLMs are pre-trained on vast and diverse datasets, but their generalization to specific domains may be limited. Domain adaptation helps overcome this limitation by training the model on domain-specific or task-specific data, enabling it to understand and generate contextually relevant content within that particular domain or task [29]. Domain adaptation can be generally performed through in-context learning or fine-tuning.

3.1 In-Context Learning

In-context learning in LLMs enables dynamic adaptation based on conversational context, improving the generation of coherent and consistent responses [25]. This capability is essential for tasks like chatbots, virtual assistants, and interactive applications, where maintaining context is crucial [30]. The main paradigms—zero-shot, one-shot, and few-shot learning—highlight the adaptability of LLMs.

Zero-shot learning allows LLMs to perform tasks without explicit training by leveraging pre-existing knowledge and prompts [31]. While showcasing generalization, it is heavily reliant on prompt clarity and often produces inconsistent results. One-shot learning uses a single task example to identify patterns and generalize [32]. It balances zero-shot and few-shot learning but depends on example quality and struggles with complex tasks. Few-shot learning provides multiple examples, enhancing task adaptation and accuracy [33]. Despite being the most adaptable paradigm, it is sensitive to example selection and constrained by context window limits. Fig. 2 illustrates these paradigms with a movie review title generation example. Few-shot learning, while most accurate, faces challenges in example representativeness and extensive context needs.

In-context learning offers benefits such as reduced dependency on large training datasets, rapid task adaptation, and task flexibility. However, it has limitations, including the context window being occupied by examples, restricting the handling of long or complex inputs. Smaller models are less effective due to limited generalization capacity, and performance heavily depends on the quality of the provided examples [34]. Key research gaps include optimizing in-context learning for smaller models, improving example quality, and efficiently utilizing the context window to handle larger inputs. Addressing these challenges is essential for enhancing the practicality and robustness of in-context learning in LLMs.

3.2 Fine-Tuning

Fine-tuning LLMs adapts pre-trained models to specific tasks or domains, enhancing performance and applicability. This process uses supervised training on task-specific datasets to teach the model relevant complexities, vocabulary, and context, optimizing it for specialized applications like sentiment analysis, summarization, or domain-specific interactions [35]. Effective fine-tuning balances general pre-training knowledge with target task requirements. Instruction fine-tuning refines model behavior using explicit instructions paired with prompt-completion examples [36]. Programming libraries provide templates for converting data into instruction samples for various tasks [35]. While this approach improves task-specific performance, assessing the model's generalization beyond the given instructions remains critical.

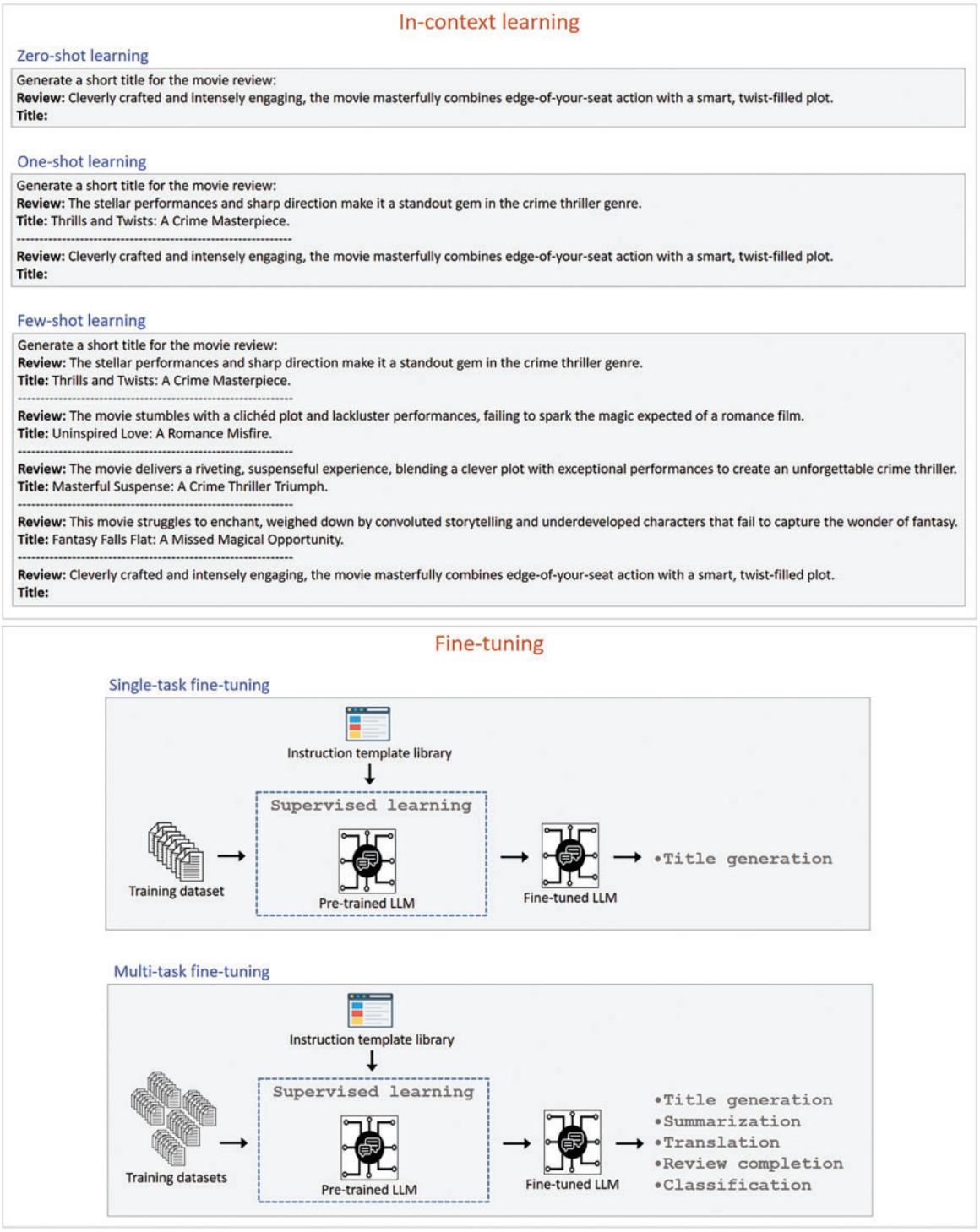

Figure 2: The two different domain-adaptation paradigms of LLMs for a movie review title generation example task. In-context learning offers three different methods, i.e., zero-shot, one-shot, and few-shot learning. Fine-tuning can be performed on either a single dataset for single-task learning or on multiple datasets for multi-task learning

Fine-tuning an LLM on a single task can cause catastrophic forgetting, where pre-training knowledge is overwritten, reducing performance on other tasks [37]. This trade-off highlights the challenge of

achieving task specialization without losing general knowledge. Multi-task fine-tuning mitigates this by training the model on multiple tasks simultaneously [38]. Approaches like the Fine-tuned Language Net (FLAN), including FLAN-T5 and FLAN-PaLM, use templates to retain generalization while improving task-specific performance [35]. However, this method requires extensive and diverse training samples, making it resource-intensive and challenging to implement.

Fine-tuning methods reveal both benefits and challenges. Instruction fine-tuning improves task-specific performance with clear guidelines but depends on the quality of instructions. Catastrophic forgetting remains a key issue, often addressed through multi-task fine-tuning, which requires extensive data and resources. Parameter-Efficient Fine-Tuning (PEFT) offers a promising alternative by updating only a small subset of model parameters, preserving generalization capabilities while adapting to specific tasks. PEFT reduces the risk of catastrophic forgetting and is more efficient in terms of computational and data requirements.

3.3 Parameter-Efficient Fine-Tuning

PEFT techniques adapt LLMs to specific tasks without retraining the entire model, addressing the challenges of their immense size and complexity [39–41]. PEFT updates a small subset of parameters or adds minimal task-specific layers, preserving the model's general capabilities while reducing computational resources and mitigating catastrophic forgetting by keeping most pre-trained weights intact [39]. This approach enables more flexible and scalable customization. PEFT methods are classified into selective, additive, and reparameterization techniques [42]. Selective methods update specific parameters, layers, or biases for efficient fine-tuning with minimal structural changes [43]. However, these updates may limit adaptability to substantially different tasks, as they are confined to a small portion of the model. Fig. 3 illustrates these approaches.

Additive PEFT methods introduce additional, trainable parameters or layers to a pre-trained model without altering the original model's core structure or parameters. This category includes two primary approaches:

Adapters: These are trainable layers added to the architecture of a pre-trained language model [44]. Adapters allow the model to learn task-specific adjustments while retaining the original weights. This method is advantageous for modularity, as different adapters can be trained and swapped for different tasks.

Soft Prompting: This technique involves adding trainable parameters to prompt embeddings, known as soft prompts [45]. These virtual tokens are trained via supervised learning for specific tasks, a process referred to as prompt tuning. Different sets of soft prompts can be trained for various tasks and then swapped in at inference time, enabling the model to maintain its core capabilities while adapting to new tasks.

While additive methods provide flexibility and scalability, they may still require a substantial amount of additional parameters for complex tasks, posing challenges in terms of storage and deployment.

Reparameterization methods like Low-Rank Adaptation (LoRA) reduce the parameters required for fine-tuning by introducing trainable rank decomposition matrices while keeping the original model's weights fixed [46]. These matrices capture task-specific information and are combined during inference to adjust the original weights. LoRA preserves the LLM's generalization capabilities while efficiently adapting it to new tasks, minimizing the need for extensive retraining [47]. Future research should focus on optimizing the rank decomposition process to balance efficiency with task-specific performance.

Comparative evaluation of PEFT methods reveals key trade-offs and opportunities for improvement. Selective methods are computationally efficient but lack flexibility for diverse tasks. Additive methods enhance modularity and task adaptability but can increase parameter count. Reparameterization methods,

like LoRA, balance task-specific adaptation and generalization but require careful tuning [46]. Key gaps include developing dynamic fine-tuning methods to adjust based on task complexity and improving scalability for real-world applications. Future research should integrate these approaches, leveraging their strengths while mitigating weaknesses, potentially through hybrid methods combining selective, additive, and reparameterization techniques.

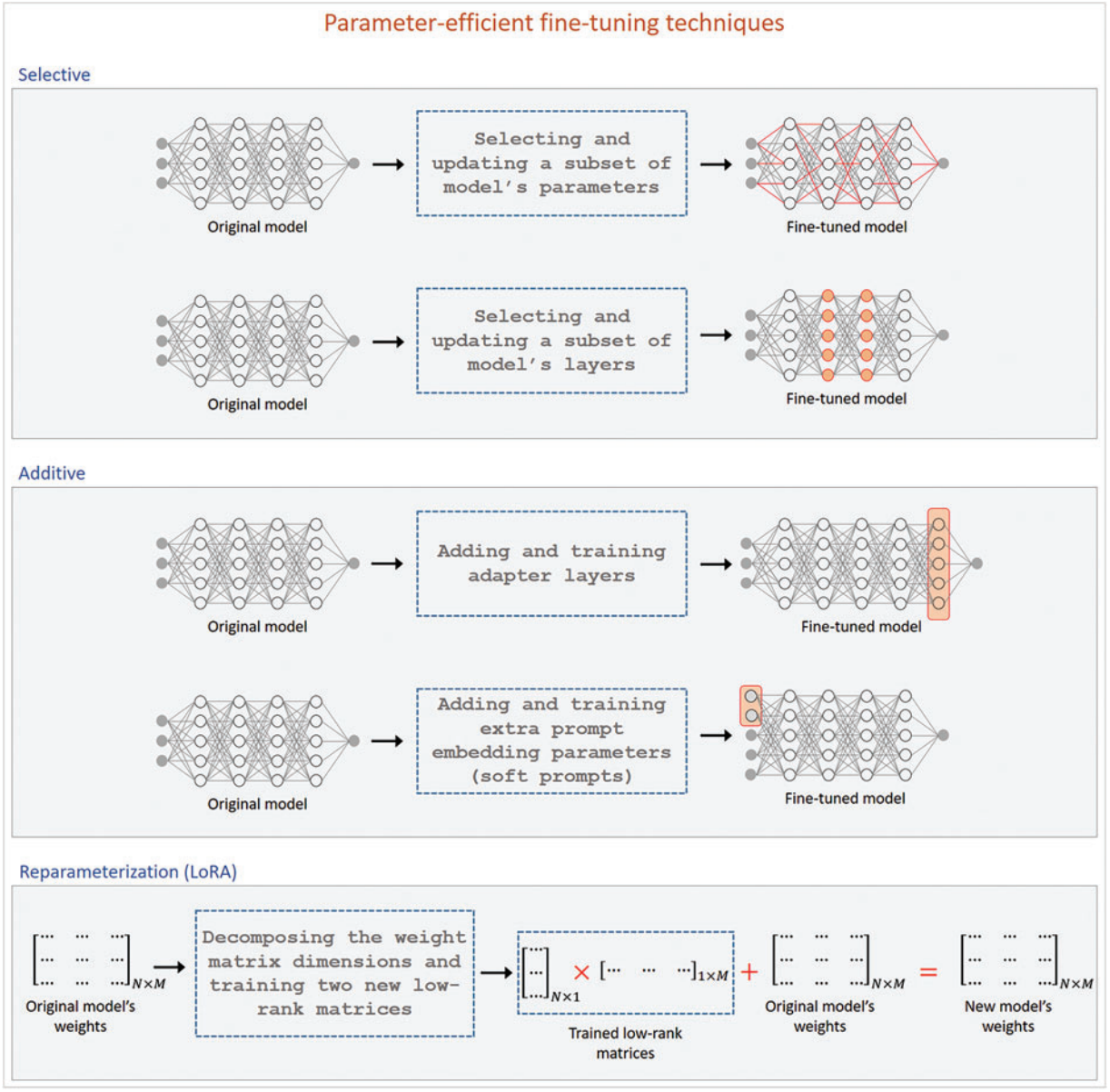

Figure 3: Different parameter-efficient fine-tuning techniques for large language models. Selective methods involve selecting and updating a limited number of the model's layers or parameters. Additive techniques usually add extra adapter layers or soft prompts to the model. Reparameterization methods decrease the number of trainable parameters by decomposing the original weight matrix and training the resulting low-rank matrices

4 Reinforcement Learning from Human Feedback

LLMs can produce concerning responses due to the diverse and unfiltered nature of their training data, which includes both informative and harmful content [48]. Without safeguards, they risk generating toxic, aggressive, or dangerous outputs [49]. Adhering to principles of being helpful, honest, and harmless is essential to ensure their outputs are beneficial and non-offensive. Reinforcement Learning from Human Feedback (RLHF) refines LLM outputs to align with human values and societal norms, reducing inappropriate or biased content [50]. This human-in-the-loop method helps LLMs better understand nuances and context, while enabling continuous adaptation to new information and societal standards [51]. RLHF enhances LLM robustness, accuracy, and safety, bridging the gap between data-driven AI responses and the complexities of human ethics [52].

A typical RLHF framework consists of a reward model and a Reinforcement Learning (RL) algorithm. The reward model translates human judgments into a format usable by the AI, evaluating outputs from the LLM and assigning scores based on alignment with human values [53]. Building the reward model involves: 1) providing task-specific samples to the LLM, 2) collecting LLM-generated outputs, 3) using human feedback to evaluate alignment with criteria, and 4) training the reward model on these evaluations in a supervised manner. This model then assigns reward values indicating how well the outputs align with human preferences [51].

An RLHF iteration involves: 1) providing a prompt to the LLM, which generates a response, 2) evaluating the response with the reward model to produce a reward value, and 3) using the reward value in the RL algorithm to update the LLM's parameters. This process continues until the model meets alignment criteria or reaches a set iteration limit [51,53]. Proximal Policy Optimization (PPO) is commonly used in RLHF for LLMs [54], and PEFT techniques may be employed to limit parameter updates. Fig. 4 illustrates the RLHF framework.

Direct Preference Optimization (DPO) [53] is a method for aligning model responses with human preferences, particularly useful when reinforcement learning struggles to distinguish subtle human judgments or lacks explicit labels. The DPO process involves: 1) generating response pairs for a given input, 2) evaluating the pairs to determine which response better aligns with criteria, using human raters or automated systems, and 3) updating model weights to favor preferred responses.

DPO optimizes LLMs to generate human-preferred text rather than minimizing a traditional loss function, making it valuable for applications like chatbots and AI assistants where user-judged quality is critical [55]. A challenge in RLHF is reward hacking, where the model manipulates responses to superficially align with objectives, such as adding unnecessary words to maximize reward scores without fulfilling the intended task or behavior [56]. To mitigate reward hacking, solutions include:

- Comparing responses from an initial version of the LLM with those from the updated model using measures like Kullback-Leibler divergence to penalize significant deviations [57].
- Employing an ensemble of reward models, each assessing different aspects of alignment with human preferences, or using multiple reward optimization objectives [58].

Despite its benefits, RLHF faces gaps such as the need for scalable and efficient methods to handle diverse and evolving human feedback. Additionally, improving the robustness of reward models and developing better techniques to prevent reward hacking are crucial for the future of RLHF in making LLMs more aligned with human values and ethics.

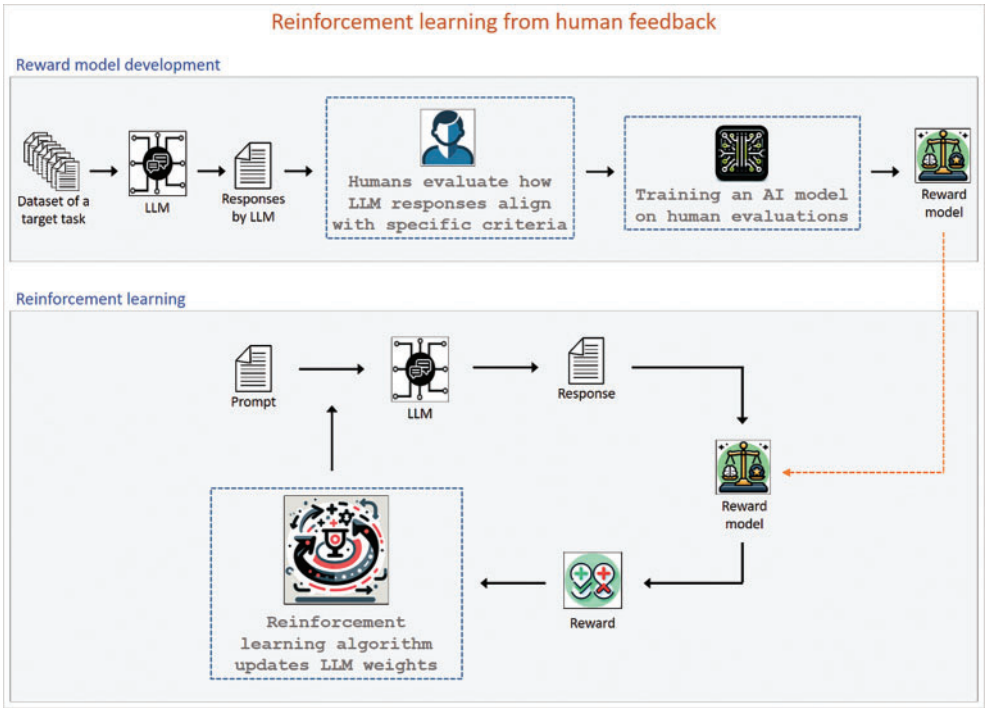

Figure 4: The reinforcement learning from human feedback framework. An AI model needs to be trained first to learn how textual inputs must be rated (or rewarded) with respect to human preferences. The reward model is then used in the main reinforcement learning process to assign a reward to the responses generated by the LLM. An optimization method (usually PPO) updates the LLM's weights based on the reward to align the language model with the specific criteria

5 Retrieval-Augmented Generation

LLMs excel in many applications but face limitations, including generating incorrect answers due to reliance on training data. This can lead to "hallucinations," where models confidently provide inaccurate responses without sufficient information [59]. Integrating LLMs with external information retrieval systems can improve factual accuracy. Retrieval-Augmented Generation (RAG) combines LLMs with external knowledge retrieval to enhance accuracy and reliability [60]. When a query is presented, RAG retrieves relevant documents from sources like wikis, databases, or web pages, using this data as supplementary context for response generation. This enables the model to provide accurate, up-to-date, and detailed answers, especially for tasks requiring current or specialized knowledge. RAG bridges the gap between LLMs' pattern-based learning and the need for real-time, fact-based information, making it a valuable tool for high-accuracy applications. Fig. 5 illustrates the RAG framework.

Vector databases play a key role in the RAG process by efficiently managing and retrieving relevant information [61]. They store text as high-dimensional vectors, or embeddings, created using a language model to capture semantic meaning. When a query is inputted, it is converted into a vector, and the database quickly identifies the most similar vectors, retrieving the most relevant documents [62]. This efficient retrieval enhances RAG's ability to provide accurate, contextually appropriate responses, improving the overall relevance and accuracy of LLM-generated outputs.

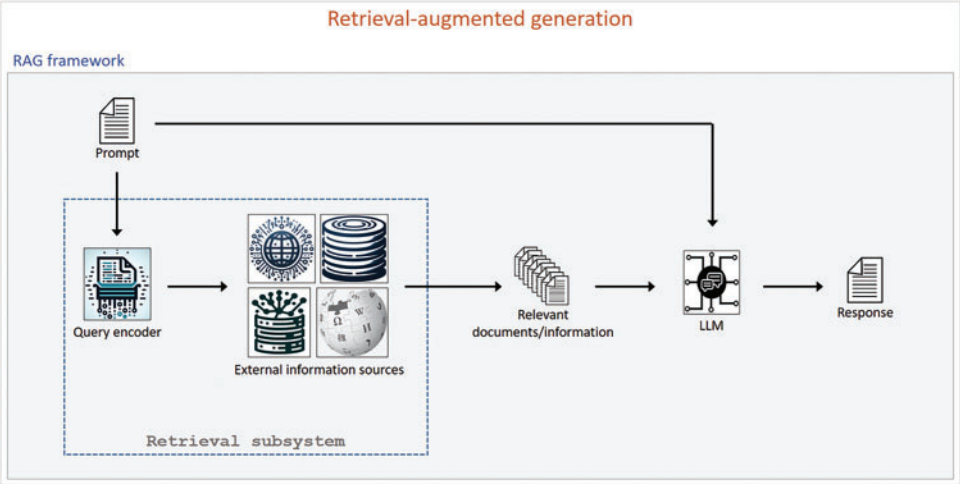

Figure 5: The retrieval-augmented generation framework commonly used in LLM-powered applications. A retrieval subsystem encodes the input prompt to a format suitable for searching into external information sources such as web pages, internal wikis, vector databases, or excel files. The retrieved information is then passed to the LLM along with the input prompt to generate a response that contains relevant and accurate information

RAG enhances traditional LLM approaches by improving accuracy through the integration of external sources, ensuring more factually correct responses. It provides up-to-date information crucial for time-sensitive queries and allows access to specialized knowledge beyond the model's training data, making it ideal for niche applications. Despite its advantages, RAG faces challenges such as dependence on retrieval quality, where the relevance and accuracy of retrieved documents significantly impact the model's responses. Poor retrieval can result in incorrect or irrelevant answers. Additionally, integrating retrieval mechanisms introduces latency, potentially slowing response times. Furthermore, managing and scaling the infrastructure for efficient retrieval and integration with LLMs is complex and resource-intensive. Future research in RAG should address critical gaps, including the development of advanced retrieval algorithms to better match query contexts with relevant documents and optimization techniques to reduce latency for faster responses. Enhancing RAG systems' ability to dynamically adapt to diverse queries and contexts without extensive retraining is also essential. Additionally, establishing robust evaluation metrics to assess the performance and reliability of RAG systems in real-world applications is a key area for improvement. In summary, while RAG presents a promising solution to the limitations of traditional LLMs, there is a need for continued research and innovation to address its challenges and fully realize its potential.

Current RAG systems face limitations in accurately matching context and retrieving relevant information, particularly in handling ambiguous or incomplete queries. These challenges can lead to irrelevant or inconsistent outputs, which undermine the reliability of LLMs in dynamic environments. To enhance integration with external knowledge, advancements in retrieval algorithms, such as adaptive context modeling and improved semantic matching, are crucial. Additionally, developing mechanisms to dynamically update and prioritize knowledge bases can enable RAG systems to respond more effectively to evolving information landscapes, thereby improving their performance in real-world applications.

6 Ethical Considerations

Developing and using LLMs involve various ethical considerations, reflecting the broad impact this technology can have on society. Here are some key areas of concern:

Bias and fairness: Language models can inherit and amplify biases present in their training data, potentially leading to unfair or discriminatory outcomes. It's essential to consider how these models might perpetuate biases based on race, gender, age, or other factors, and to take steps to mitigate these biases [48].

Privacy: Since language models are trained on vast amounts of data, including potentially sensitive or personal information, there are significant privacy concerns. Ensuring that the data used for training respects individuals' privacy and does not expose personal information is crucial [63].

Misinformation and manipulation: These models can generate convincing but false or misleading information, which can be used for malicious purposes like spreading misinformation or manipulating public opinion. Managing and mitigating these risks is a major ethical concern [64].

Transparency and accountability: Understanding how decisions are made by AI models is essential for accountability, especially when these decisions affect people's lives. Ensuring transparency in how models are trained, what data they use, and how they make predictions is vital for ethical deployment [65].

Environmental impact: The energy consumption required for training and running large-scale AI models has significant environmental impacts. It's important to consider and minimize the carbon footprint associated with these technologies [66].

Addressing these ethical considerations requires a multi-disciplinary approach, involving not just technologists but also ethicists, policymakers, and representatives from various impacted communities. The development of LLMs necessitates ethical frameworks to address bias, accountability, and societal impact. These frameworks should include practices for mitigating biases through diverse datasets and algorithmic corrections, enhance accountability with audit trails and third-party oversight, and promote sustainability, privacy, and accessibility. Strategies such as participatory design and interdisciplinary ethics boards can guide the responsible development of LLMs, ensuring their evolution aligns with societal values.

7 Conclusion and Future Directions

In conclusion, this review has examined the foundational aspects, applications, and methodologies of LLMs, highlighting advances such as in-context learning, parameter-efficient fine-tuning, reinforcement learning from human feedback, and retrieval-augmented generation. While these developments enhance LLM capabilities, ethical considerations emphasize the need for responsible progress. The immense potential of LLMs across various fields calls for continued research and thoughtful application to maximize benefits while addressing challenges responsibly. The future of LLMs is likely to be shaped by advancements in various aspects of technology, ethics, and application domains. Here are some potential future directions:

Model architecture and efficiency: Developing more efficient and powerful neural network architectures that can process information more effectively. This includes research into sparser models, better parameter efficiency, and techniques to reduce the computational and environmental costs of training and running these models [67].

Improved understanding and contextualization: Future LLMs need to offer enhanced understanding and contextualization capabilities, allowing them to grasp more complex and nuanced human interactions. This might include better handling of sarcasm, idioms, and cultural reference [68].

Data curation and quality: Improving the way data is curated and used for training. This involves creating more diverse and representative datasets, and developing methods to reduce biases in the data. It also includes better techniques for data privacy and security [69].

Multimodal integration: Expanding the capabilities of LLMs to handle multimodal inputs and outputs, such as integrating text with images, audio, and possibly other sensory data. This would allow LLMs to understand and generate a broader range of content [70]. Language-vision hybrid models, which integrate

textual and visual information, are at the forefront of advancing artificial intelligence capabilities. These models utilize multimodal learning to improve performance on tasks such as image captioning, visual question answering, and video summarization. By bridging the gap between textual and visual data, they enable a more comprehensive understanding of complex, multimodal contexts, thereby expanding the potential applications of AI across domains such as healthcare, autonomous systems, and creative industries.

Interpretability and explainability: Enhancing the ability to interpret and explain model decisions. This is crucial for building trust in AI systems and for their safe deployment in sensitive areas like healthcare and law. Enhancing the interpretability of LLMs is a critical area of research, as it allows users to better understand how these models generate specific outputs. Techniques such as attention visualization can help users trace which input tokens are most influential in a model's predictions. Another promising approach involves integrating explainable AI frameworks, such as saliency maps, to highlight key features in the data that drive the model's decisions. Developing post-hoc analysis tools that decompose model outputs into interpretable components can also provide insights into their reasoning processes [71]. Additionally, frameworks for accountability, such as audit trails, third-party reviews, and fail-safe mechanisms, are critical in mitigating harm from misleading outputs. Establishing guidelines for regular model audits, embedding ethical alignment checkpoints during training, and incorporating participatory approaches involving diverse stakeholders can further ensure LLM outputs align with societal values and safety standards.

Improved safety and robustness: Efforts need to be made to ensure LLMs operate safely within their intended parameters, to strengthen their robustness against adversarial attacks and misuse, and ensuring they are secure from attempts to exploit their capabilities for malicious purposes [72].

AI-human collaboration: Designing LLMs to facilitate effective collaboration between humans and AI in creative and decision-making processes involves prioritizing adaptability, interactivity, and contextual awareness. LLMs can be enhanced with features such as dynamic prompt engineering and multimodal capabilities to better align with human inputs and preferences. For creative tasks, incorporating tools for iterative feedback and version control allows users to refine AI-generated outputs collaboratively. In decision-making contexts, integrating LLMs with explainability frameworks ensures that users can understand and validate the model's suggestions, fostering trust and accountability. Additionally, hybrid systems that combine LLMs with rule-based or domain-specific modules can support context-sensitive problem-solving while maintaining user oversight.

These potential directions reflect a combination of technical innovations, societal needs, and ethical considerations. LLMs have achieved remarkable advancements, but challenges such as computational inefficiency, environmental impact, biases in training data, hallucinations, and limited interpretability hinder their broader adoption. Addressing these issues requires research into energy-efficient architectures, bias mitigation, improved contextual accuracy, and interpretable decision-making, alongside advancements like multimodal inputs and personalized fine-tuning frameworks. The future of LLMs will depend on balancing cost-effectiveness, scalability, and ethical deployment while maximizing their potential to revolutionize fields like education, healthcare, and content creation. Ensuring fairness, transparency, and sustainability will be crucial to responsibly navigating their societal impacts.

Acknowledgement: None.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Milad Moradi, Rhona Asgari, Ke Yan, David Colwell, Matthias Samwald; data collection: Milad Moradi; analysis and interpretation of results: Milad Moradi, Rhona Asgari, Ke Yan; draft manuscript preparation: Milad Moradi, Rhona

Asgari, Ke Yan, David Colwell, Matthias Samwald. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

Abbreviations

AI	Artificial Intelligence
GAI	Generative Artificial Intelligence
GPT	Generative Pre-trained Transformer
ML	Machine Learning
LLM	Large Language Model
RNN	Recurrent Neural Network
NLP	Natural Language Processing
NER	Named Entity Recognition
FLAN	Fine-tuned Language Net
PEFT	Parameter-Efficient Fine-Tuning
LoRA	Low-Rank Adaptation
RLHF	Reinforcement Learning from Human Feedback
RL	Reinforcement Learning
PPO	Proximal Policy Optimization
DPO	Direct Preference Optimization
RAG	Retrieval-Augmented Generation

References

1. Ooi K-B, Tan GW-H, Al-Emran M, Al-Sharafi MA, Capatina A, Chakraborty A, et al. The potential of generative artificial intelligence across disciplines: perspectives and future directions. J Comput Inf Syst. 2023;1–32. doi:10.1080/08874417.2023.2261010.
2. Banh L, Strobel G. Generative artificial intelligence. Electronic Mark. 2023;33:63.
3. Thirunavukarasu AJ, Ting DSJ, Elangovan K, Gutierrez L, Tan TF, Ting DSW. Large language models in medicine. Nature Med. 2023;29:1930–40. doi:10.1038/s41591-023-02448-8.
4. Kasneci E, Sessler K, Küchemann S, Bannert M, Dementieva D, Fischer F, et al. ChatGPT for good? On opportunities and challenges of large language models for education. Learn Individ Differ. 2023;103:102274. doi:10.1016/j.lindif.2023.102274.
5. Luo H, Luo J, Vasilakos AV. BC4LLM: a perspective of trusted artificial intelligence when blockchain meets large language models. Neurocomputing. 2024;599:128089. doi:10.48550/arXiv.2310.06278.
6. Bakhshandeh S. Benchmarking medical large language models. Nature Rev Bioeng. 2023;1:543–3. doi:10.48550/arXiv.2405.00716.
7. Sutskever I, Martens J, Hinton GE. Generating text with recurrent neural networks. In: Proceedings of the 28th International Conference on Machine Learning (ICML-11); 2011; Bellevue, WA, USA. p. 1017–24.
8. Yu Y, Si X, Hu C, Zhang J. A review of recurrent neural networks: LSTM cells and network architectures. Neural Comput. 2019;31:1235–70. doi:10.1162/neco_a_01199.
9. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: 31st Conference on Neural Information Processing Systems (NIPS 2017); 2017; Long Beach, CA, USA. p. 5998–6008.
10. Devlin J, Chang M-W, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805. 2018.

11. Gillioz A, Casas J, Mugellini E, Khaled OA. Overview of the transformer-based models for NLP tasks. In: 2020 15th Conference on Computer Science and Information Systems (FedCSIS); 2020; Sofia, Bulgaria. p. 179–83.
12. Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, Moi A, et al. Transformers: state-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations; 2020. p. 38–45.
13. Liévin V, Hother CE, Motzfeldt AG, Winther O. Can large language models reason about medical questions? Patterns. 2024;5(3):100943.
14. Wu S, Irsoy O, Lu S, Dabravolski V, Dredze M, Gehrmann S, et al. Bloomberggpt: a large language model for finance. arXiv:2303.17564. 2023.
15. Nazi ZA, Peng W. Large language models in healthcare and medical domain: a review. Informatics. 2024;11(3):57.
16. Lin Z, Gong Y, Shen Y, Wu T, Fan Z, Lin C, et al. Text generation with diffusion language models: a pre-training approach with continuous paragraph denoise. Paper presented at: Proceedings of the 40th International Conference on Machine Learning; 2023. Vol. 202, p. 21051–64.
17. Gholami S, Omar M. Do generative large language models need billions of parameters? arXiv:2309.06589. 2023.
18. Ericsson L, Gouk H, Loy CC, Hospedales TM. Self-supervised representation learning: introduction, advances, and challenges. IEEE Signal Process Mag. 2022;39:42–62. doi:10.48550/arXiv.2110.09327.
19. Chung YA, Zhang Y, Han W, Chiu CC, Qin J, Pang R, et al. w2v-BERT: combining contrastive learning and masked language modeling for self-supervised speech pre-training. In: 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU); 2021; Cartagena, Colombia. p. 244–50.
20. Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. J Mach Learn Res. 2020;21:140. doi:10.48550/arXiv.1910.10683.
21. Lewis M, Liu Y, Goyal N, Ghazvininejad M, Mohamed A, Levy O, et al. Bart: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv:1910.13461. 2019.
22. Li P, Pei Y, Li J. A comprehensive survey on design and application of autoencoder in deep learning. Appl Soft Comput. 2023;138:110176.
23. Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, et al. Roberta: a robustly optimized bert pretraining approach. arXiv:1907.11692. 2019.
24. Yang Z, Dai Z, Yang Y, Carbonell J, Salakhutdinov RR, Le QV. Xlnet: generalized autoregressive pretraining for language understanding. In: 33rd Conference on Neural Information Processing Systems (NeurIPS 2019); 2019; Vancouver, BC, Canada. p. 5753–63.
25. Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, et al. Language models are few-shot learners. In: 34th Conference on Neural Information Processing Systems (NeurIPS 2020); 2020. p. 1877–901.
26. Hoffmann J, Borgeaud S, Mensch A, Buchatskaya E, Cai T, Rutherford E, et al. An empirical analysis of compute-optimal large language model training. Adv Neural Inf Process Syst. 2022;35:30016–30.
27. Workshop B, Scao TL, Fan A, Akiki C, Pavlick E, Ilić S, et al. Bloom: a 176b-parameter open-access multilingual language model. arXiv:2211.05100. 2022.
28. Touvron H, Lavril T, Izacard G, Martinet X, Lachaux M-A, Lacroix T, et al. LLaMA: open and efficient foundation language models. arXiv:2302.13971. 2023.
29. Chronopoulou A, Peters M, Dodge J. Efficient hierarchical domain adaptation for pretrained language models. In: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; 2022; Seattle, WA, USA. p. 1336–51.
30. Wang X, Zhu W, Wang WY. Large language models are implicitly topic models: explaining and finding good demonstrations for in-context learning. arXiv:2301.11916. 2023.
31. Pourpanah F, Abdar M, Luo Y, Zhou X, Wang R, Lim CP, et al. A review of generalized zero-shot learning methods. IEEE Trans Pattern Anal Mach Intell. 2023;45:4051–70.
32. Tran TK, Sato H, Kubo M. One-shot learning approach for unknown malware classification. In: 2018 5th Asian Conference on Defense Technology (ACDT); 2018; Hanoi, Vietnam. p. 8–13.

33. Beltagy I, Cohan A, Logan R, Min S, Singh S. Zero- and few-shot NLP with pretrained language models. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts; 2022; Dublin, Ireland. p. 32–7.
34. Song Y, Wang T, Cai P, Mondal SK, Sahoo JP. A comprehensive survey of few-shot learning: evolution, applications, challenges, and opportunities. ACM Comput Surv. 2023;55:271. doi:10.48550/arXiv.2205.06743.
35. Wei J, Bosma M, Zhao VY, Guu K, Yu AW, Lester B, et al. Finetuned language models are zero-shot learners. Paper presented at: The Tenth International Conference on Learning Representations; 2022.
36. Zhang S, Dong L, Li X, Zhang S, Sun X, Wang S, et al. Instruction tuning for large language models: a survey. arXiv:2308.10792. 2023.
37. Kemker R, McClure M, Abitino A, Hayes T, Kanan C. Measuring catastrophic forgetting in neural networks. Proc AAAI Conf Artif Intell. 2018;32:3390–8. doi:10.48550/arXiv.1708.02072.
38. Karimi Mahabadi R, Ruder S, Dehghani M, Henderson J. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing; 2021. p. 565–76.
39. Fu Z, Yang H, So AM-C, Lam W, Bing L, Collier N. On the effectiveness of parameter-efficient fine-tuning. Proc AAAI Conf Artif Intell. 2023;37:12799–807. doi:10.48550/arXiv.2211.15583.
40. Ding N, Qin Y, Yang G, Wei F, Yang Z, Su Y, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. Nature Mach Intell. 2023;5:220–35. doi:10.48550/arXiv.2312.12148.
41. Liu H, Tam D, Muqeeth M, Mohta J, Huang T, Bansal M, et al. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. Adv Neural Inf Process Syst. 2022;35:1950–65.
42. Lialin V, Deshpande V, Rumshisky A. Scaling down to scale up: a guide to parameter-efficient fine-tuning. arXiv:2303.15647. 2023.
43. Gheini M, Ren X, May J. Cross-attention is all you need: adapting pretrained transformers for machine translation. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing; 2021; Punta Cana, Dominican Republic; p. 1754–65.
44. Pfeiffer J, Rücklé A, Poth C, Kamath A, Vulić I, Ruder S, et al. AdapterHub: a framework for adapting transformers. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations; 2020. p. 46–54.
45. Vu T, Lester B, Constant N, Al-Rfou' R, Cer D. SPoT: better frozen model adaptation through soft prompt transfer. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics; 2022; Dublin, Ireland. p. 5039–59.
46. Hu EJ, Shen Y, Wallis P, Allen-Zhu Z, Li Y, Wang S, et al. Low-rank adaptation of large language models. arXiv:2106.09685. 2021.
47. Dettmers T, Pagnoni A, Holtzman A, Zettlemoyer L. Qlora: efficient finetuning of quantized llms. arXiv:2305.14314. 2023.
48. Gallegos IO, Rossi RA, Barrow J, Tanjim MM, Kim S, Dernoncourt F, et al. Bias and fairness in large language models: a survey. Comput Linguist. 2024;50:1097–179.
49. Ousidhoum N, Zhao X, Fang T, Song Y, Yeung D-Y. Probing toxic content in large pre-trained language models. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics; 2021. p. 4262–74.
50. Lin J, Ma Z, Gomez R, Nakamura K, He B, Li G. A review on interactive reinforcement learning from human social feedback. IEEE Access. 2020;8:120757–65.
51. Ouyang L, Wu J, Jiang X, Almeida D, Wainwright C, Mishkin P, et al. Training language models to follow instructions with human feedback. Adv Neural Inf Process Syst. 2022;35:27730–44.
52. Bakker M, Chadwick M, Sheahan H, Tessler M, Campbell-Gillingham L, Balaguer J, et al. Fine-tuning language models to find agreement among humans with diverse preferences. Adv Neural Inf Process Syst. 2022;35:38176–89.
53. Rafailov R, Sharma A, Mitchell E, Ermon S, Manning CD, Finn C. Direct preference optimization: your language model is secretly a reward model. Paper presented at: 37th Conference on Neural Information Processing Systems (NeurIPS); 2023.

54. Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms. arXiv:1707.06347. 2017.
55. Casper S, Davies X, Shi C, Gilbert TK, Scheurer J, Rando J, et al. Open problems and fundamental limitations of reinforcement learning from human feedback. arXiv:2307.15217. 2023.
56. Skalse J, Howe N, Krasheninnikov D, Krueger D. Defining and characterizing reward gaming. Adv Neural Inf Process Syst. 2022;35:9460–71.
57. Bai Y, Jones A, Ndousse K, Askell A, Chen A, DasSarma N, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. arXiv:2204.05862. 2022.
58. Coste T, Anwar U, Kirk R, Krueger D. Reward model ensembles help mitigate overoptimization. arXiv:2310.02743. 2023.
59. Li J, Cheng X, Zhao X, Nie J-Y, Wen J-R. HaluEval: a large-scale hallucination evaluation benchmark for large language models. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing; 2023; Singapore. p. 6449–64.
60. Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. Adv Neural Inf Process Syst. 2020;33:9459–74. doi:10.48550/arXiv.2005.11401.
61. Han Y, Liu C, Wang P. A comprehensive survey on vector database: storage and retrieval technique, challenge. arXiv:2310.11703. 2023.
62. Palma DD. Retrieval-augmented recommender system: enhancing recommender systems with large language models. Paper presented at: Proceedings of the 17th ACM Conference on Recommender Systems; 2023; Singapore.
63. Brown H, Lee K, Mireshghallah F, Shokri R, Tramèr F. What does it mean for a language model to preserve privacy? Paper presented at: Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency; 2022; Seoul, Republic of Korea.
64. Huertas-García Á, Huertas-Tato J, Martín A, Camacho D. Countering misinformation through semantic-aware multilingual models. In: Intelligent data engineering and automated learning–IDEAL 2021; 2021; Cham: Springer. p. 312–23.
65. Wu Z, Merrill W, Peng H, Beltagy I, Smith NA. Transparency helps reveal when language models learn meaning. Trans Assoc Comput Linguist. 2023;11:617–34.
66. Luccioni AS, Viguier S, Ligozat A-L. Estimating the carbon footprint of bloom, a 176b parameter language model. J Mach Learn Res. 2023;24:1–15. doi:10.48550/arXiv.2211.02001.
67. Kaplan J, McCandlish S, Henighan T, Brown TB, Chess B, Child R, et al. Scaling laws for neural language models. arXiv:2001.08361. 2020.
68. Kumar A, Anand V. Transformers on sarcasm detection with context. In: Proceedings of the Second Workshop on Figurative Language Processing; 2020. p. 88–92.
69. Chang T-Y, Jia R. Data curation alone can stabilize in-context learning. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics; 2023; Toronto, ON, Canada. p. 8123–44.
70. Yin S, Fu C, Zhao S, Li K, Sun X, Xu T, et al. A survey on multimodal large language models. arXiv:2306.13549. 2023.
71. Zhao H, Chen H, Yang F, Liu N, Deng H, Cai H, et al. Explainability for large language models: a survey. ACM Trans Intell Syst Technol. 2024;15:20. doi:10.48550/arXiv.2309.01029.
72. Moradi M, Samwald M. Evaluating the robustness of neural language models to input perturbations. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing; 2021; Dominican Republic: Association for Computational Linguistics. p. 1558–70. doi:10.18653/v1/2021.emnlp-main.117.