A Survey on Large Language Models and Their Integration into System Architectures

www.surveyx.cn

Abstract

This survey paper explores the multifaceted integration of Large Language Models (LLMs) into system architectures, emphasizing their transformative impact across various domains, including healthcare, education, and diplomacy. LLMs enhance natural language processing (NLP) capabilities, enabling more intuitive human-machine interactions and facilitating complex tasks such as optimization and automated diagnostics. The survey identifies key challenges, such as computational inefficiencies and integration complexities, and examines strategies like federated learning and edge device deployment to mitigate these issues. It highlights the role of operating systems in resource management and the importance of innovative scheduling frameworks for optimizing AI agent performance. Additionally, the paper discusses the architectural patterns and frameworks that support the scalability and security of AI systems, underscoring the need for adaptable and efficient designs. Future research opportunities are outlined, focusing on improving LLM robustness, ethical considerations in AI deployment, and the development of advanced methodologies for enhancing AI integration into diverse applications. By providing a comprehensive overview of current trends and challenges, this survey aims to inform ongoing research and development efforts, ensuring the responsible and effective deployment of LLMs in AI-driven technologies.

1 Introduction

1.1 Importance of Large Language Models

Large Language Models (LLMs) are pivotal in modern technological advancement, offering transformative capabilities across numerous industries. Their integration with physiological data enhances empathy in human-computer interactions by accurately interpreting users' mental and emotional states, addressing limitations in existing methods [1]. In computing education, LLMs have significantly influenced educational technology, providing innovative avenues for teaching and learning through generative models [2].

Despite their strengths, LLMs face challenges in generalization, particularly in automating complex tasks such as web browsing and gaming, necessitating ongoing research to enhance their applicability [3]. In the optimization domain, LLMs are being integrated with optimization algorithms, improving decision-making and efficiency in solving complex problems [4].

In healthcare, LLMs like ChatGPT are explored for their potential in automated and cross-modal diagnostics, notably in fields such as dentistry [5]. Their application in diplomacy highlights the necessity for advanced negotiation and social reasoning skills in complex multi-agent interactions [6].

Challenges persist, particularly regarding inefficiencies in serving engines during multi-turn conversations, which affect cost-effectiveness and performance [7]. Moreover, integrating LLMs with operating systems is crucial for enhancing security, reliability, and performance, underscoring the need for adaptable system architectures [8].

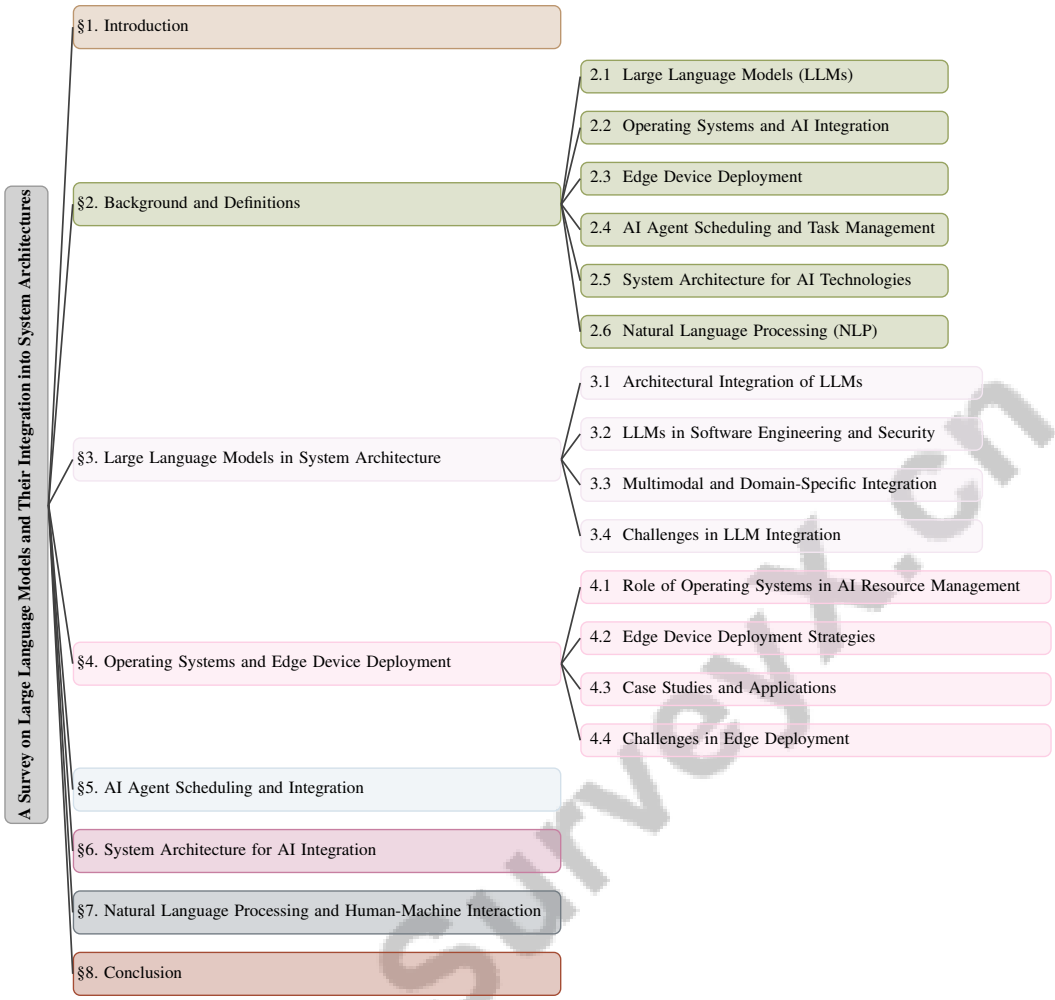


Figure 1: chapter structure

The significance of LLMs extends beyond innovation; they enhance processes such as scientific literature reviews, machine translation in crises, and the development of responsible AI applications. Recent studies indicate that LLMs, especially those based on the GPT architecture, are revolutionizing systematic reviews, improving data extraction accuracy, and enabling tailored multilingual translation systems for low-resource languages during emergencies. Their evolving integration necessitates a focus on transparency and responsible deployment to address the unique challenges they present [9, 10, 11, 12]. Their impact on healthcare, education, and diplomacy underscores the need for continued exploration and optimization of their capabilities alongside addressing deployment challenges.

1.2 Scope and Objectives

This survey provides a comprehensive analysis of integrating Large Language Models (LLMs) into system architectures, focusing on diverse applications, inherent challenges, and potential solutions. A key objective is to assess the impact of LLMs on occupational dynamics, particularly job displacement in China, highlighting the need for adaptive strategies across industries [13]. The survey explores extensive applications of LLMs in fields such as natural language processing, education, and healthcare, drawing insights from a broad range of research, including 194 papers related to ChatGPT [14].

The integration of LLMs with Knowledge Graphs (KGs) will be examined to enhance knowledge representation and address challenges like hallucinations in AI outputs [15]. The survey evaluates methods like CachedAttention, aiming to reduce computational overheads in LLMs, thereby improving efficiency and cost-effectiveness [7]. Additionally, the potential of LLMs to enhance

problem-solving therapy sessions and autonomous vehicle systems will be explored, emphasizing their role in processing complex natural language requests.

The scope includes evaluating detectors designed to identify harmful outputs from LLMs, contributing to the development of safer and more reliable AI systems [16]. The integration of LLMs in optimization tasks will also be investigated to refine their capabilities and expand practical applications [4]. Furthermore, the challenges of developing AI systems that exhibit human-like creativity will be addressed, focusing on evaluating and improving such systems [17].

The survey aims to provide an in-depth analysis of the current landscape and future potential of integrating LLMs into system architectures, examining diverse applications, challenges, and methodologies associated with LLM deployment while considering various stakeholders' perspectives and the importance of transparency in fostering responsible AI development [18, 10, 11]. This will contribute to advancements in AI-driven technologies and their applications across various domains, ensuring effective deployment of LLMs aligned with industry needs and societal expectations.

1.3 Relevance to Current Trends

The integration of Large Language Models (LLMs) into various technological domains increasingly aligns with contemporary trends across industries. A significant aspect of this alignment is enhancing user interfaces through LLMs, reflecting the ongoing focus on improving user experience and functionality via AI technologies. Frameworks like Prompt Sapper demonstrate how LLMs enable users to interact with AI using natural language, broadening access to AI capabilities [19]. This trend is exemplified by the OS-Copilot framework, facilitating generalist computer agents to learn and adapt to various tasks within operating systems, underscoring the development of versatile AI systems capable of handling diverse tasks [3].

In creative coding, LLMs augment human-AI collaboration, reflecting the growing role of AI in artistic and creative processes [2]. Their ability to enhance multilingual capabilities, particularly for low-resource languages, aligns with increasing demand for inclusive and diverse AI applications [20]. In education, LLM integration is revolutionizing teaching practices and prompting a reevaluation of academic integrity policies, marking a significant shift in higher education [14]. This transformation is part of a broader trend toward leveraging AI to enhance learning outcomes and methodologies.

Moreover, the integration of LLMs into business processes emphasizes the importance of trust in AI, as these technologies evolve and integrate into critical operations. Findings from studies on smaller language models, optimized through the CMAT framework, highlight the potential for smaller models in real-world applications, suggesting a shift toward more efficient and scalable AI solutions [21]. Similarly, the compressor-retriever architecture provides a model-agnostic solution for managing context across sessions by efficiently compressing and retrieving information, aligning with trends in optimizing AI models for better performance [22].

In cybersecurity, integrating LLMs into penetration testing aligns with current trends in automation, providing a standardized evaluation framework for AI models [23]. Additionally, integrating asynchronous AI agents addresses the limitations of current AI systems operating in a strict turn-based fashion, enhancing multitasking and interactivity [24]. The application of LLMs in autonomous vehicles supports current trends in AI development, particularly in enhancing human-vehicle communication and decision-making capabilities [25].

Despite these advancements, challenges persist regarding the reliability of LLMs in providing accurate information, as evidenced by their performance in areas like SP advice [26]. The integration of LLMs into multimodal road network modeling aligns with current trends in AI applications, enhancing user interaction and improving the efficiency of complex systems [27]. The integration of LLMs is advancing technological capabilities while fostering a deeper understanding of AI's societal role, aligning with the current emphasis on ethical and informed AI development. Additionally, addressing inefficiencies and high resource demands associated with training large language models further aligns with current technological trends [28].

1.4 Structure of the Survey

This survey systematically explores the multifaceted integration of Large Language Models (LLMs) into system architectures. The paper is divided into key sections, each focusing on distinct aspects of LLM integration and its implications.

The survey begins with the **Introduction**, which underscores the importance of LLMs and outlines the scope, objectives, and relevance of the study in the context of current technological trends. This section sets the stage for a detailed exploration of how LLMs are reshaping various domains.

Following the introduction, the **Background and Definitions** section provides a comprehensive overview of essential concepts and technologies relevant to the survey. It defines LLMs, operating systems, edge device deployment, AI agent scheduling, AI integration, system architecture, and natural language processing, establishing a foundational understanding necessary for subsequent discussions.

The third section, **Large Language Models in System Architecture**, delves into the role of LLMs within system architectures, examining methods for their integration and the challenges that arise. This includes discussions on architectural integration, applications in software engineering and security, and domain-specific considerations.

In **Operating Systems and Edge Device Deployment**, the survey explores how operating systems manage resources for AI applications, particularly in edge device deployment. This section highlights strategies for deploying AI models on edge devices, supported by case studies and applications, while also addressing the challenges faced in this area.

The fifth section, **AI Agent Scheduling and Integration**, focuses on optimizing task management and resource allocation through effective AI agent scheduling. It discusses challenges, strategies for optimization, and innovative scheduling frameworks, providing insights into seamless AI integration into existing systems.

System Architecture for AI Integration analyzes the structural design of systems incorporating AI technologies, emphasizing scalability, flexibility, and performance. This section discusses architectural patterns, scalability techniques, security measures, and emerging trends in AI system design.

The penultimate section, **Natural Language Processing and Human-Machine Interaction**, explores advancements in NLP through LLM integration and its role in enhancing human-machine interaction. It highlights applications across diverse contexts, showcasing the transformative impact of NLP technologies.

Finally, the **Conclusion** summarizes key findings and reflects on future directions and research opportunities in integrating LLMs into system architectures. This section provides a forward-looking perspective, emphasizing the potential for further advancements and innovations in AI-driven technologies.The following sections are organized as shown in Figure 1.

2 Background and Definitions

2.1 Large Language Models (LLMs)

Large Language Models (LLMs) mark a pivotal advancement in artificial intelligence, particularly in natural language processing (NLP). Utilizing extensive datasets and sophisticated architectures, LLMs achieve near-human proficiency in tasks such as literature review automation, decision-making optimization, and data augmentation, thereby transforming research methodologies and practical applications [29, 10, 9, 4]. Their vast parameter counts enable them to generate and comprehend text with human-like fluency, raising questions about their ability to convey meaningful content akin to traditional linguistic constructs.

LLMs automate complex tasks, such as regulatory content extraction and document compliance verification, bridging regulatory analysis with technological innovation [30]. They excel in text summarization, distilling extensive texts into concise summaries while retaining essential information [31]. Moreover, they empower non-experts to prototype machine learning functionalities without extensive technical expertise, broadening access to AI technologies [32].

In specialized domains, LLMs enhance large-scale data processing efficiencies [33]. Their integration with knowledge graphs mitigates hallucinations by embedding structured knowledge [15]. However, challenges persist in effectively learning to utilize external tools for complex tasks, necessitating dynamic learning capabilities [34].

LLMs are also employed in cybersecurity, notably in honeypot systems that interact with attackers to bolster security measures [35]. Robust detectors are essential to identify harmful content, ensuring safe and ethical deployment [16]. Despite their potential, LLMs struggle to discern users' mental and emotional states, crucial for applications like psychotherapy [1]. Their limitations in generalizing across applications and handling real-time feedback pose challenges for efficient task execution [3].

In translation, LLMs demonstrate capabilities in both supervised and zero-shot translations, contingent on parallel data availability [36]. The exploration of LLMs' creative potential continues, aiming to enhance their ability to generate creative content [17]. Optimizing graph flattening processes is crucial for improving LLM performance in understanding long-range dependencies in complex data environments [37].

As LLMs evolve, ongoing research is vital to address limitations, optimize deployment, and ensure safe integration into diverse applications. This evolution underscores LLMs' roles in shaping AI's future across education, research, and societal applications, raising concerns about transparency, academic integrity, and responsible AI deployment [38, 39, 9, 2, 11].

2.2 Operating Systems and AI Integration

Operating systems (OS) are crucial for integrating and managing AI resources, facilitating interactions between hardware and software. Advanced paradigms like AIOS position LLMs as core OS intelligence, optimizing resource allocation and enhancing user interfaces through natural language processing. This integration streamlines operations and improves system performance while addressing privacy and security challenges [40, 41, 42, 43, 44].

A significant challenge is the absence of a universal OS for cognitive robots that adapts to various platforms and tasks, necessitating a versatile architecture that dynamically adjusts to different environmental configurations. Enhanced adaptability is crucial for deploying AI-based systems effectively, particularly in diverse hardware and software contexts, as studies examine the impacts of OS and architectures on AI outputs. Understanding AI technologies' interplay with OS functions is imperative for achieving optimal resource utilization and reliability [45, 46, 40, 47, 48].

Current robotic OS often lack a data-driven architecture, essential for rapid AI application development and deployment. Integrating generalist computer agents highlights the need for advanced resource management capabilities supporting diverse applications and facilitating seamless user-AI system interactions. Addressing challenges like background noise interference and user input variability is crucial for enhancing speech-enabled systems' accuracy and reliability across applications from education to mobile computing [39, 49].

Incorporating new heterogeneous technologies into existing OS often requires substantial modifications for optimal performance. Traditional systems are increasingly barriers to efficiently utilizing specialized processors in massively parallel computations due to outdated file structures. Innovative OS like TabulaROSA and ColonyOS better manage resources across heterogeneous environments, leveraging mathematical frameworks and microservice architectures to optimize performance [45, 50, 51, 43]. This necessitates new OS architectures supporting complex workflows and data interactions in scientific and engineering applications.

Moreover, integrating generative AI models into OS facilitates natural language interactions, enabling more intuitive user interfaces. This advancement requires enhancements in AI and machine learning applications within the OS kernel, particularly in memory management, process scheduling, and I/O operations. However, developers face challenges in writing eBPF programs due to kernel complexities and limitations imposed by the eBPF verifier, complicating kernel functionality extension [8].

The role of OS in AI integration is multifaceted, involving adaptations of traditional architectures to meet modern AI demands. Effective AI system deployment requires enhancing resource management capabilities, integrating heterogeneous technologies, and facilitating natural language interactions. These elements optimize LLMs like ChatGPT, which demonstrate significant potential across ap-

plications such as text summarization and language translation. Understanding their computational demands and leveraging innovative integrations are essential for maximizing their impact [31, 38].

2.3 Edge Device Deployment

Edge device deployment involves running AI models on local devices, such as smartphones and IoT devices, rather than relying solely on centralized cloud servers. This approach enhances privacy, stability, and personalization in AI applications. Deploying LLMs and Large Multimodal Models (LMMs) on mobile devices exemplifies this trend, enabling users to process data locally while maintaining control over personal information [52].

Integrating AI models into edge devices addresses the critical need for reduced latency and improved operational efficiency. Edge computing architectures utilizing Docker and Kubernetes enhance Unmanned Aerial Vehicles' (UAVs) computational capabilities, significantly reducing delays and boosting performance [53]. Such architectures facilitate real-time data processing and decision-making, essential for applications like autonomous vehicles and smart city infrastructures.

The performance of LLMs on mobile platforms is assessed through benchmarks focusing on user experience metrics and hardware dynamics, ensuring that model deployment does not compromise device functionality or efficiency [54]. As demand for AI-driven applications on personal devices grows, models must be both powerful and resource-efficient.

Federated fine-tuning emerges as a pivotal strategy for optimizing LLMs on edge devices, allowing models to be refined using data from multiple devices without transferring it to a central server. This enhances energy efficiency and complies with emerging AI regulations, such as the EU AI Act [55]. Federated learning (FL) solutions address challenges in deploying AI across heterogeneous device ecosystems, including communication efficiency, client heterogeneity, and privacy concerns [56].

2.4 AI Agent Scheduling and Task Management

AI agent scheduling and task management are crucial for efficiently deploying AI systems, particularly when leveraging LLMs. A primary challenge is sub-optimal scheduling and resource allocation of agent requests, leading to bottlenecks and inefficient utilization of LLM capabilities [44]. This inefficiency is exacerbated by current AI agents' inability to handle multiple concurrent processes, resulting in delays and poor user experiences during long-running tasks [24].

Integrating LLMs into task management systems requires careful consideration of their unique features and overcoming implementation barriers related to data handling and evaluation [57]. Selecting the appropriate LLM for specific applications is complex, involving an understanding of the diverse functionalities and limitations of available models. Furthermore, distilling LLMs into smaller language models (SLMs) often inherits flawed reasoning and hallucinations, complicating task management and necessitating strategies to mitigate these issues [58].

To optimize task management, innovative scheduling frameworks are required to facilitate seamless execution of tasks across multiple agents. These frameworks must enable asynchronous processing and real-time tool usage, enhancing overall efficiency and user experience [24]. Additionally, insights from human-AI collaboration, such as those from artists working with LLMs, can inform the optimization of task management strategies [59].

2.5 System Architecture for AI Technologies

Integrating AI technologies into system architectures necessitates a multifaceted approach addressing technical and sociotechnical challenges. A key aspect is optimizing resource allocation to support large-scale AI deployments requiring substantial computational resources. Frameworks like Infinite-LLM, which utilize pooled GPU memory across clusters, exemplify strategies to enhance scalability and efficiency [60]. Such frameworks are essential for managing the heavy computational loads associated with LLMs, ensuring that system architectures can effectively support their training and execution.

Advanced parallelism techniques are integral to designing system architectures for AI technologies. Methods such as PTD-P, combining pipeline, tensor, and data parallelism, are instrumental in handling

LLMs' computational demands [61]. These techniques enable efficient workload distribution across multiple processing units, enhancing performance and reducing latency.

Flexibility and adaptability in system architectures are crucial for managing heterogeneous resources. The LLaMaS framework, which uses LLMs as an OS module to interpret device characteristics, exemplifies this adaptability [43]. This aligns with the broader trend of developing architectures that dynamically adjust to varying technological requirements and environmental configurations.

Innovations in hardware architecture, such as silicon photonics, significantly enhance computational capabilities and energy efficiency [54]. Additionally, frameworks like TabulaROSA, employing associative array algebra for managing OS functions in massively parallel compute engines, provide robust solutions for AI integration [50].

Sociotechnical challenges, such as ensuring the reliability and trustworthiness of AI outputs, are integral to designing system architectures. The taxonomy of AIOps tasks illustrates how LLMs can enhance data preprocessing, failure perception, root cause analysis, and auto-remediation by providing accurate insights [62]. Ethical considerations in UI/UX design, such as avoiding deceptive patterns in writing assistants, are essential for ensuring transparency and user-friendliness [63].

Inference optimization techniques are crucial in system architecture design. Frameworks categorizing these techniques emphasize the importance of hardware-aware strategies to optimize model performance, ensuring efficient operation across platforms [64]. Specialized training models like OWL, focusing on IT operations data, demonstrate the potential for contextually relevant outputs unattainable by general models [65].

The structural design of system architectures for AI technologies involves integrating advanced frameworks, optimization techniques, and innovative hardware solutions. Addressing technical and ethical challenges enables effective incorporation of AI technologies, enhancing functionality, adaptability, and trustworthiness across domains. A survey of existing research on LLMs categorizes areas such as performance assessment, instructional approaches, ethical implications, and teaching material generation, highlighting the comprehensive scope of AI integration into system architectures [2]. Frameworks like Mélange, which determine minimal-cost GPU allocation for LLM services while meeting service-level objectives, underscore the importance of cost-efficient resource management in AI system design [66]. As hardware scaling trends reveal diminishing returns due to communication overhead, the need for innovative architectural solutions becomes increasingly apparent [45].

2.6 Natural Language Processing (NLP)

Natural Language Processing (NLP) is a critical domain within artificial intelligence focused on enabling computers to understand, interpret, and generate human language. This field employs computational techniques to analyze and synthesize natural language and speech, facilitating meaningful interactions between humans and machines. The integration of LLMs into NLP systems has significantly expanded their capabilities, enhancing fluency and coherence in processing natural language [67].

Advancements in NLP are evident across applications, including machine translation, where extensive datasets comprising millions of sentences ensure robust evaluation across multiple languages [36]. This capability is crucial in crisis communication contexts, where accurate and rapid language processing can be lifesaving [12]. Furthermore, NLP techniques are instrumental in data augmentation processes, organized into categories such as data creation, labeling, reformation, and co-annotation, each enhancing data quality [29].

Despite these advancements, challenges remain in ensuring the contextual appropriateness and reliability of responses generated by NLP systems. Inadequacies in systems like ChatGPT can lead to misinformation, highlighting the need for continual improvements in NLP technologies [68]. Additionally, addressing biases in NLP applications, particularly in sensitive areas like recruitment and education, is crucial for ensuring equitable outcomes [69].

NLP systems also enhance privacy protection capabilities, as demonstrated by benchmarks progressively testing models for privacy awareness [70]. The performance of NLP systems in creative tasks is assessed using metrics like the CREATIVITY INDEX, quantifying originality based on n-gram rarity and pushing the boundaries of novel content generation [71].

In social intelligence, NLP evaluates multi-turn interactions across diverse scenarios, providing insights into AI models' social capabilities [72]. Moreover, NLP techniques enable LLMs to manipulate human psychology and organizational dynamics, enhancing communication strategies such as phishing emails [67].

NLP is a transformative field that continues to evolve, driven by advanced AI model integration and a commitment to addressing challenges related to accuracy, fairness, and creativity. The significance of LLMs in enhancing human-machine interaction is substantial, facilitating intuitive, efficient, and meaningful communication across diverse applications, including writing assistance, language translation, and accessibility tools for individuals with disabilities. Ongoing development raises important considerations regarding transparency, creativity, and ethical design, underscoring the need for responsible integration into everyday interactions [38, 63, 71, 11, 49].

In recent years, the application of Large Language Models (LLMs) has gained significant attention in various fields, particularly in system architecture and software engineering. The integration of LLMs into architectural frameworks not only enhances functionality but also presents unique challenges that must be addressed for effective implementation. As illustrated in Figure 2, this figure provides a comprehensive overview of the integration of LLMs within system architecture. It outlines their application across different architectural frameworks, highlights their role in multimodal and domain-specific contexts, and delineates the challenges faced during their integration. This visual representation serves to enhance our understanding of the multifaceted nature of LLMs and their impact on contemporary system design.

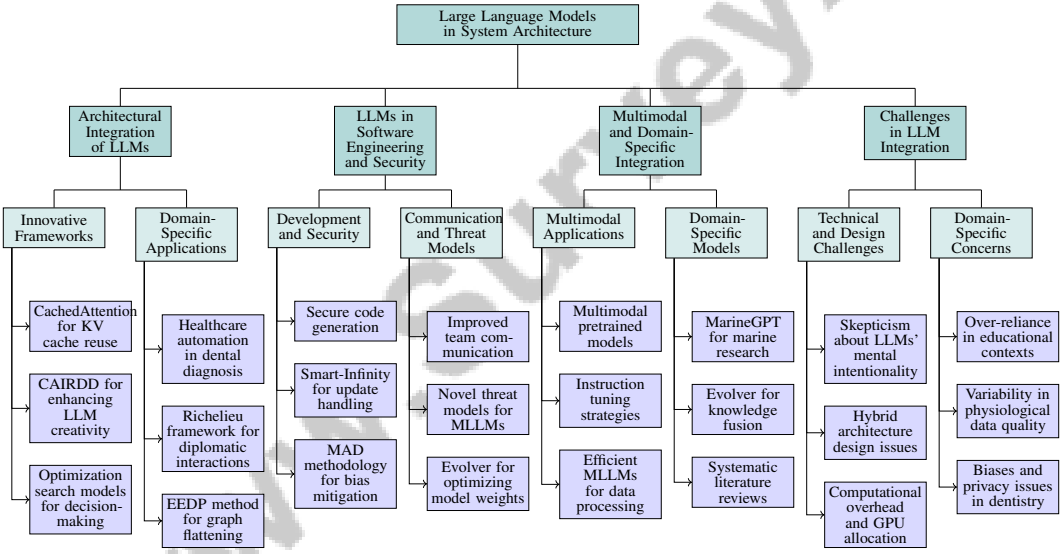


Figure 2: This figure illustrates the integration of Large Language Models (LLMs) in system architecture, outlining their application across architectural frameworks, software engineering, multimodal and domain-specific contexts, and the challenges faced in their integration.

3 Large Language Models in System Architecture

3.1 Architectural Integration of LLMs

Integrating Large Language Models (LLMs) into system architectures involves employing innovative frameworks that enhance functionality and efficiency across diverse domains. CachedAttention exemplifies this by optimizing key-value (KV) cache reuse across conversation turns, reducing computational overhead in real-time environments [7]. In creative applications, the CAIRDD method enhances LLM creativity by generating and refining concepts based on user-defined criteria [17]. LLMs also function as optimization search models, enhancing decision-making and operational efficiency [4]. In healthcare, LLMs automate dental diagnosis and treatment planning by integrating diverse data sources, improving accuracy and efficiency [5].

Method Name	Optimization Techniques	Application Domains	Automation and Efficiency
CA[7]	Cachedattention Innovates	Multi-turn Conversations	Improving Operational Efficiency
CAIRDD[17]	Iterative Process	Creative Applications	Improve Operational Efficiency
MMLLM[5]	Multi-modal Data	Dental Diagnosis	Treatment Planning
R[6]	Self-evolving Mechanism	Diplomatic Contexts	Strategic Planning
EEDP[37]	Path Compression	-	Improving Operational Efficiency

Table 1: This table presents a comparative analysis of various methods integrating Large Language Models (LLMs) with a focus on optimization techniques, application domains, and automation efficiency. The methods highlighted include CachedAttention, CAIRDD, MMLLM, Richelieu, and EEDP, each demonstrating unique contributions to enhancing operational efficiency and strategic planning across diverse fields such as multi-turn conversations, creative applications, dental diagnosis, diplomatic contexts, and graph flattening. The table underscores the versatility and impact of LLMs in advancing AI technologies across multiple domains.

The Richelieu framework showcases self-evolving LLM-based agents for complex diplomatic interactions, emphasizing strategic planning and negotiation [6]. Additionally, the End-to-End DAG-Path prompting (EEDP) method optimizes graph flattening processes based on human cognition [37]. This integration is crucial for improving transparency, automating processes in literature reviews and legal compliance, and ensuring responsible AI deployment [30, 10, 11].

As illustrated in Figure 3, the integration of LLMs into various architectural frameworks highlights optimization techniques, application domains, and automation in reviews. This figure showcases the diversity and impact of LLMs across different sectors such as healthcare, diplomacy, and legal compliance. Table 1 provides a detailed overview of the integration of Large Language Models (LLMs) into various architectural frameworks, highlighting the optimization techniques employed, the application domains served, and the resulting improvements in automation and efficiency. As LLMs evolve, their integration into various applications will play a pivotal role in advancing AI technologies across multiple domains.

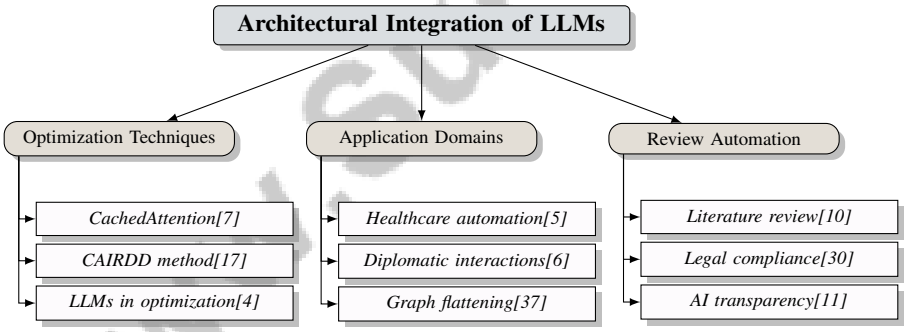


Figure 3: This figure illustrates the integration of Large Language Models (LLMs) into various architectural frameworks, highlighting optimization techniques, application domains, and automation in reviews. It showcases the diversity and impact of LLMs across different sectors such as healthcare, diplomacy, and legal compliance.

3.2 LLMs in Software Engineering and Security

The application of Large Language Models (LLMs) in software engineering significantly influences development processes and security practices. LLMs excel at generating secure code and addressing vulnerabilities, enhancing software robustness [73]. Decoder-only LLMs dominate software engineering tasks, surpassing other architectures like encoder-only and encoder-decoder models [57]. Addressing harmful, biased, or privacy-violating content is crucial, with techniques like machine unlearning mitigating these issues [74]. Smart-Infinity optimizes update handling through custom accelerators in computational storage devices, enhancing software development efficiency [75].

LLMs also improve communication quality among development teams, facilitating clearer interactions [76]. The MAD methodology promotes exploration of diverse solutions, mitigating biases and enhancing inclusivity [77]. Novel threat models for Multimodal Large Language Models (MLLMs)

provide comprehensive frameworks for risk assessment and mitigation [78]. Evolutionary strategies like Evolver optimize model weights, enhancing LLM security and performance [79].

The integration of LLMs enhances software engineering processes, improving efficiency and security while necessitating a reassessment of traditional practices. This shift is evident in legal compliance automation, where LLMs streamline regulatory requirement extraction and analysis [30, 10, 73]. As LLMs evolve, their impact on software engineering and security practices will likely expand, driving further innovation.

3.3 Multimodal and Domain-Specific Integration

The integration of Large Language Models (LLMs) into multimodal applications and domain-specific contexts marks a significant advancement in AI, enabling comprehensive processing of diverse data types. Multimodal pretrained models enhance LLM capabilities, allowing content processing across text, images, and other data forms [80]. Instruction tuning strategies optimize interactions between modalities, improving performance across complex tasks. In domain-specific applications, models like MarineGPT generate specialized responses tailored to fields such as marine research, supporting conservation efforts and public engagement [81].

Efficient Multimodal Large Language Models (MLLMs) contribute to improved data processing and decision-making capabilities across various domains [82]. Innovative methods like Evolver, which aggregates and evolves weights from multiple fine-tuned models, exemplify advanced knowledge fusion techniques, enhancing LLM adaptability and precision [79]. The integration of LLMs into multimodal and domain-specific applications underscores their transformative potential, offering enhanced capabilities across diverse fields. As LLMs advance, their integration into specialized domains is expected to grow, enhancing processes like systematic literature reviews and driving innovation [38, 10].

3.4 Challenges in LLM Integration

Integrating Large Language Models (LLMs) into various systems presents numerous challenges that hinder seamless deployment and effective utilization. Skepticism regarding LLMs' mental intentionality affects trust in their outputs, raising questions about authenticity and reliability [83]. The design of hybrid architectures struggles with integrating pretrained models from diverse architectures without retraining [84]. Computational overhead increases linearly with conversation turns due to inefficiencies in handling KV caches [7], compounded by suboptimal GPU allocation [66]. Frameworks like OSCAR highlight the need for systems adaptable to real-time feedback and task-driven re-planning [3].

In educational contexts, LLM integration raises concerns about over-reliance on AI tools and academic integrity [2]. Variability in physiological data quality impacts LLM accuracy, necessitating robust data collection methods [1]. Evaluating LLMs trained solely on parallel data is challenging, as iterative training processes may not accurately reflect capabilities [36]. In specialized fields like dentistry, biases in training data and data privacy concerns hinder integration [5].

The vast decision space in diplomacy presents challenges inadequately addressed by current AI methods [6]. The Dynamic Parallel Processing Algorithm (DPPA) dynamically adjusts processing strategies based on dataset size, addressing performance bottlenecks [33]. EEDP improves reasoning capabilities in long-distance scenarios, surpassing traditional methods [37], but integrating such methods into existing systems requires substantial modifications.

Addressing these challenges is crucial for maximizing LLM potential and ensuring successful integration into diverse systems. Ongoing research should focus on overcoming limitations and enhancing LLM capabilities for diverse applications, particularly in areas like systematic reviews where they automate key stages. Ensuring transparency in LLM deployment is essential for responsible AI development, necessitating a human-centered approach that considers stakeholder needs and the unique challenges posed by LLM-infused applications [10, 11].

4 Operating Systems and Edge Device Deployment

Deploying AI applications on edge devices demands intricate resource management and operational strategies within operating systems. As AI technologies become more pervasive, operating systems must evolve to meet the specific challenges of edge deployments, necessitating frameworks that optimize resource utilization and enhance AI application performance. The following subsection examines the critical role of operating systems in AI resource management.

4.1 Role of Operating Systems in AI Resource Management

Operating systems are pivotal in managing resources for AI applications, ensuring efficiency and optimizing computational resource utilization across various environments. Integrating AI components into operating systems requires robust frameworks to handle the complex workflows associated with AI deployments. The KEN framework exemplifies this by simplifying the writing of eBPF programs through natural language inputs, making kernel extensions accessible to non-experts while maintaining high correctness rates [8]. This underscores the importance of user-friendly interfaces in enhancing operating systems' functionality for AI resource management.

The substantial computational resource requirements of Large Language Models (LLMs) pose a notable challenge, necessitating effective resource management. The HypergraphOS framework addresses this by managing resources for algorithms like the Dynamic Parallel Processing Algorithm (DPPA), which requires real-time dataset analysis [33]. Such frameworks enable precise resource allocation, ensuring efficient AI operation.

Training and optimizing LLMs further highlight the need for advanced resource management strategies. Techniques like CAIRDD enhance LLM-generated content's creativity by optimizing computational resource use through iterative processes of user input, random idea generation, and evaluation [17]. Additionally, integrating optimization algorithms into operating systems addresses computational resource challenges, enhancing AI deployment efficiency [4].

Moreover, the End-to-End DAG-Path prompting (EEDP) method improves model comprehension of complex graph relationships by aligning graph representation processes with cognitive strategies, optimizing resource management in AI applications [37]. This approach emphasizes aligning computational strategies with cognitive processes to enhance resource utilization efficiency.

Operating systems are integral to the successful deployment and management of AI applications, providing the infrastructure necessary for optimizing resource utilization, enhancing performance, and ensuring compliance with ethical and legal standards. As AI technologies evolve, operating systems' role in resource management will become increasingly significant, fostering innovation and efficiency in AI-driven applications across various fields. Emerging paradigms, such as integrating large generative AI models into operating systems, will facilitate more intuitive human-computer interactions, allowing users to communicate naturally with their devices. Furthermore, resource-aware machine learning techniques are being explored to optimize core operating system functions in low-resource environments, enhancing trustworthiness and performance. The development of specialized operating systems like AIOS, which embed large language models to optimize resource allocation and manage multiple intelligent agents, marks a critical advancement. However, these innovations present challenges related to transparency, privacy, and ethical use, necessitating robust safeguards to protect user data and ensure responsible AI technology deployment [40, 41, 11, 44].

4.2 Edge Device Deployment Strategies

Deploying AI models on edge devices involves strategic considerations to enhance processing efficiency and minimize latency, crucial for real-time applications. Optical operations have emerged as a promising method, utilizing matrix-vector multiplications and logic gate implementations to accelerate neural network processing on edge devices [85]. This method leverages optical computing's inherent parallelism, providing significant speed advantages over traditional electronic systems.

Frameworks like MobileAIBench facilitate deploying Large Language Models (LLMs) and Large Multimodal Models (LMMs) on mobile platforms by evaluating quantization and resource utilization impacts on model performance [52]. This evaluation is vital for optimizing model deployment,

ensuring mobile device resource constraints do not compromise AI application functionality or efficiency.

Strategies for deploying control applications on edge devices, such as Unmanned Aerial Vehicles (UAVs), have been explored using containerization technologies like Docker and Kubernetes. These technologies enhance resource management and reduce latency, improving operational efficiency for edge-deployed AI models [53]. Such architectures are particularly beneficial for applications requiring immediate data processing and decision-making capabilities.

Benchmarking efforts quantify the variability induced by different environment configurations in AI-based systems, guiding practitioners in selecting optimal deployment configurations [47]. This approach ensures AI models are deployed under conditions that maximize performance while minimizing resource consumption.

Advanced deployment strategies also include federated fine-tuning, employing adaptive learning rates and decoupled weight decay to stabilize training and enhance convergence speed in federated settings [55]. This method is crucial for edge device deployment, allowing models to be refined with distributed data from multiple devices while preserving privacy and minimizing communication overhead.

The Any-Precision LLM method further optimizes edge deployment by reducing memory overhead and deployment costs, making sophisticated AI models feasible on resource-constrained devices [86]. This aligns with the trend of developing cost-efficient, scalable AI solutions.

Moreover, exploring additional parallelization methods significantly impacts communication efficiency and model performance on edge devices [45]. These methods are essential for optimizing AI model deployment, ensuring effective operation across diverse platforms and environments.

Deploying AI models on edge devices necessitates a multifaceted approach, combining advanced computational techniques, resource management strategies, and innovative deployment frameworks. These strategies are crucial for optimizing AI application performance, particularly in real-time and resource-constrained environments, enhancing efficiency, scalability, and responsiveness. For example, advancements in document-wise memory architecture and document guidance loss in LLMs improve the retrieval and generation of contextually relevant information, increasing models' effectiveness in dynamic scenarios. Additionally, lightweight detection methods for AI-generated text, such as ZigZag ResNet, demonstrate how computational efficiency can be achieved without sacrificing accuracy, enabling AI technologies to be deployed in various applications despite limited resources [87, 38, 9].

4.3 Case Studies and Applications

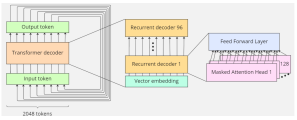
The deployment of AI models on edge devices has been exemplified through successful applications across multiple industries, showcasing the effectiveness of edge computing strategies. One notable example is the CyberCortex.AI platform, evaluated in collaborative robotics applications, including a forest fire prevention system and an autonomous driving system. In the forest fire prevention application, a legged robot and a drone worked collaboratively to enhance perception and motion control, demonstrating the potential of edge-deployed AI models in improving real-time decision-making and operational efficiency in environmental monitoring [88].

Another case study involved the TabulaROSA operating architecture tested on a supercomputer with over 32,000 cores, managing more than 68 billion processes. This simulation highlighted the system's capability to efficiently handle massive parallel computations, providing insights into the scalability and performance advantages of edge computing solutions in high-performance computing environments [50].

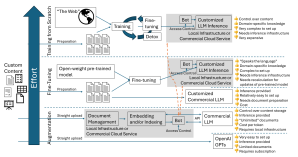
HyperGraphOS was evaluated across diverse domains, including virtual dialog systems, robotic task planning, and dynamic research projects, demonstrating its effectiveness in managing complex workflows and optimizing resource allocation in scientific and engineering applications. This underscores the potential of edge device deployments to enhance the functionality and efficiency of AI-driven systems [89].

These case studies exemplify the successful integration of AI models into edge devices across various industries, highlighting the transformative impact of edge computing in enhancing real-time

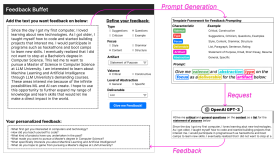
processing, scalability, and operational efficiency. As edge computing technologies advance, their integration across industries is expected to broaden significantly, facilitating enhanced innovation and optimization of AI deployment strategies. This expansion will likely leverage the capabilities of LLMs and resource-aware machine learning, increasingly utilized in low-resource environments such as the Internet of Things (IoT). As these technologies mature, they will improve operational efficiencies and introduce novel applications that could transform societal functions, necessitating a focus on responsible development and transparency to address emerging challenges [38, 71, 40, 90, 11].



(a) Transformer Recurrent Decoder Architecture[38]



(b) Effort and Custom Content in the Development of AI Systems[91]



(c) Prompt Generation and Feedback Buffet[92]

Figure 4: Examples of Case Studies and Applications

As illustrated in Figure 4, the deployment of operating systems on edge devices presents unique challenges and opportunities, as demonstrated by various case studies and applications. The "Transformer Recurrent Decoder Architecture" showcases a sophisticated structure involving a transformer decoder, recurrent decoder, and feed-forward layer, each playing a crucial role in processing and generating data tokens. This architecture exemplifies the intricate designs necessary for efficient data handling on edge devices. Additionally, the "Effort and Custom Content in the Development of AI Systems" highlights the varying levels of effort and content types required in AI system development, offering a comparative analysis of approaches such as using the web, open-weight pre-trained models, and custom content. This comparison underscores the importance of selecting the right development strategy to meet the specific needs of edge computing environments. Moreover, the "Prompt Generation and Feedback Buffet" illustrates a user-centered approach to feedback generation, utilizing an interface that leverages OpenAI's GPT-3 to refine user input through a structured feedback mechanism. Collectively, these examples underscore the diverse methodologies and innovative solutions shaping the deployment of operating systems in edge computing, enhancing edge devices' capabilities to handle complex tasks efficiently [38, 91, 92].

4.4 Challenges in Edge Deployment

Deploying AI models on edge devices presents multifaceted challenges impacting performance, scalability, and security. One significant challenge is integrating multiple Computational Storage Devices (CSDs) to ensure consistent performance across diverse hardware configurations. This complexity often results in performance bottlenecks, necessitating sophisticated orchestration strategies for effective resource management [75].

The scarcity of high-quality, domain-specific datasets for training LLMs is another critical issue complicating the fine-tuning of models for specialized tasks and increasing the risk of generating misleading outputs without proper verification [93]. Additionally, reliance on translation-based datasets for low-resource languages may fail to capture cultural nuances, exacerbating the data scarcity challenge [20].

Orchestrating multiple agents and selecting appropriate LLMs for complex environments also pose significant difficulties. Current studies often do not fully address these issues, leading to suboptimal performance and inefficiencies in task execution [94]. Moreover, integrating AI models into edge devices complicates data persistence, managing training data biases, and ensuring the security and ethical use of language models [41].

Security concerns are paramount, particularly in deploying LLMs for end-to-end penetration testing, which reveals challenges in handling exploitation and privilege escalation tasks, highlighting the need for improved methodologies and frameworks to enhance security measures [23]. Furthermore, integrating multimodal data introduces vulnerabilities to attacks, such as data poisoning, due to the complexity of image processing and the difficulty in detecting and mitigating such threats [78].

Maintaining the lightweight nature of operating systems while integrating complex machine learning algorithms presents another significant challenge. This integration must ensure system correctness and manage the unpredictability introduced by new AI methods without compromising performance [40]. Additionally, reliance on external services for security and privacy, as seen in systems like HyperGraphOS, may pose further challenges in protecting user data [89].

To effectively tackle the multifaceted challenges associated with deploying LLMs, a comprehensive strategy is essential. This strategy should encompass developing robust orchestration frameworks, enhancing dataset quality and accessibility through innovative data augmentation techniques, and implementing advanced security measures to safeguard against emerging threats, such as LLM-generated phishing attacks and harmful outputs. This approach addresses undesirable model behavior and academic integrity concerns while leveraging LLM capabilities for anomaly detection and creating tailored adversarial prompts, fostering a more secure and efficient deployment environment [39, 67, 29, 95, 96]. By addressing these issues, the deployment of AI models on edge devices can be optimized, ensuring efficient and effective operation in diverse environments.

5 AI Agent Scheduling and Integration

The progression of AI technologies necessitates adept scheduling and integration of AI agents to maximize their operational efficiency. Table 2 presents a comprehensive overview of the challenges and strategies associated with AI agent scheduling, task management optimization, and AI integration into existing systems, as discussed in the following sections. This section explores the intricate challenges associated with AI agent scheduling, particularly concerning Large Language Models (LLMs), and highlights the need for strategies that optimize their performance. Addressing these challenges is crucial for developing effective task management and resource allocation strategies, which are foundational to enhancing AI systems' capabilities.

5.1 Challenges in AI Agent Scheduling

The scheduling of AI agents, especially those employing LLMs, presents several challenges impacting their effectiveness across various applications. Key issues include latency and resource allocation inefficiencies that impede real-time performance. Existing scheduling mechanisms often inadequately utilize LLM capabilities, leading to processing delays and task execution bottlenecks [44]. Moreover, current frameworks struggle with asynchronous processing and real-time tool usage, limiting concurrent task execution and affecting user experience during prolonged operations [24]. Selecting suitable LLMs for specific tasks is further complicated by the diversity in model capabilities and task-specific requirements [57].

The challenge of flawed reasoning and hallucinations from LLMs propagating to smaller distilled models complicates task management, necessitating robust strategies to mitigate inaccuracies [58]. Additionally, integrating AI agents into existing systems often requires significant infrastructure modifications, posing compatibility and seamless integration challenges [41]. Innovative scheduling frameworks are essential to address these issues, enabling asynchronous processing and optimizing resource allocation to enhance AI system efficiency and responsiveness. Insights from human-AI collaboration, particularly in creative processes involving LLMs, can inform the development of more effective scheduling strategies [59].

5.2 Strategies for Optimizing Task Management

Enhancing task management and resource allocation for AI agents is pivotal for improving efficiency across applications. Developing asynchronous AI agents capable of processing multiple requests in real-time significantly boosts interactivity and user satisfaction by reducing latency and facilitating concurrent task execution [24]. This strategy enables AI systems to manage diverse workloads more effectively, ensuring optimal resource allocation.

Integrating advanced scheduling algorithms that prioritize tasks based on urgency and resource requirements can further improve task management. These algorithms dynamically adjust resource allocation in response to changing conditions, ensuring high-priority tasks receive necessary computational resources without sacrificing performance. By leveraging machine learning techniques, these systems continuously learn from operational data to adapt strategies for optimizing task execution,

thereby minimizing bottlenecks and enhancing responsiveness in complex environments. Additionally, incorporating LLMs can improve anomaly detection and decision-making processes, ensuring systems remain resilient and efficient amid fluctuating demands [45, 9, 40, 96, 4].

Employing modular architectures allows seamless integration of new functionalities and models, facilitating efficient resource management. These architectures enable the updating and reconfiguration of AI agents to adapt to changing task requirements, significantly enhancing adaptability and scalability by incorporating document-wise memory systems that manage relevant information throughout the learning process [10, 9, 71]. This modularity supports deploying specialized models tailored to specific tasks, ensuring optimal resource utilization.

5.3 AI Integration into Existing Systems

Seamlessly integrating AI functionalities into existing system architectures is crucial for enhancing capabilities and optimizing operations across various domains. The OS-Copilot framework exemplifies a robust approach by acting as middleware that facilitates interaction between AI agents and system components, ensuring smooth transitions and operational efficiency [97]. This framework highlights the importance of middleware solutions in bridging AI capabilities with existing infrastructures.

Integrating AI into existing systems also involves frameworks like Prompt Sapper, which embeds AI-native services directly into system architecture, leveraging underlying resources to enhance overall performance and adaptability [19]. Insights from the historical evolution of operating systems inform the development of the AIOS-Agent ecosystem, which aims to integrate AI functionalities by learning from past advancements in system architecture design and resource management [42]. This perspective underscores the need for adaptive design principles that accommodate the dynamic nature of AI technologies.

Furthermore, optimizing computational efficiency and handling diverse data sources are critical for effective AI applications within existing system constraints [62]. The incorporation of UAV control functionalities into system architectures using edge computing exemplifies how AI can enhance operational capabilities [53]. PlanFitting demonstrates another innovative approach to AI integration by generating personalized exercise plans that adapt to changing circumstances, showcasing AI's potential to deliver tailored solutions within existing frameworks [98]. This adaptability is essential for ensuring that AI systems remain relevant and effective in dynamic environments.

5.4 Innovative Scheduling Frameworks

Innovative scheduling frameworks for AI agents are crucial for optimizing resource allocation and enhancing the efficiency of AI-driven systems. The AIOS framework exemplifies this by facilitating the integration of LLMs into operating systems, enabling efficient task scheduling and management [44]. This framework addresses suboptimal resource allocation challenges by providing a structured approach to managing LLM computational and memory demands, ensuring minimal latency and maximum efficiency.

Asynchronous scheduling frameworks represent another critical innovation, allowing AI agents to process multiple tasks concurrently, thus improving system responsiveness and user satisfaction [24]. By enabling real-time tool usage and asynchronous processing, these frameworks overcome traditional scheduling limitations reliant on synchronous execution, enhancing overall AI application performance.

Moreover, advanced scheduling algorithms leveraging machine learning techniques can dynamically adjust resource allocation based on task priorities and system conditions. These algorithms learn from historical data to predict future resource demands, optimizing task execution and reducing processing bottlenecks [4]. This adaptability ensures AI systems can respond effectively to changing workloads while maintaining high efficiency across diverse applications.

Incorporating modular architectures in scheduling frameworks supports flexible integration of new AI functionalities and models. This modularity facilitates seamless updating and reconfiguration of AI agents, allowing them to adapt to evolving task requirements and leverage the latest advancements in AI technologies [62]. By enabling the integration of specialized models tailored to specific applications, these frameworks enhance the adaptability and scalability of AI systems.

Feature	Challenges in AI Agent Scheduling	Strategies for Optimizing Task Management	AI Integration into Existing Systems
Scheduling Approach	Inefficient Utilization	Asynchronous Processing	Middleware Solutions
Integration Strategy	Significant Modifications	Modular Architectures	Ai-native Services
Resource Management	Latency Bottlenecks	Dynamic Adjustment	Computational Efficiency

Table 2: This table provides a comparative analysis of methods addressing challenges in AI agent scheduling, strategies for optimizing task management, and the integration of AI into existing systems. It highlights key features such as scheduling approaches, integration strategies, and resource management techniques, offering insights into the inefficiencies and potential solutions for enhancing AI system performance.

6 System Architecture for AI Integration

6.1 Architectural Patterns and Frameworks

Integrating Large Language Models (LLMs) into system architectures relies on adaptable, scalable, and efficient frameworks. The AutoFlow framework exemplifies this by utilizing natural language programs and reinforcement learning to automate workflow generation, thereby optimizing task execution [99]. Similarly, the Graph of Thought (GoT) framework employs a graph-based structure to enable dynamic task execution, enhancing workflow automation through interconnected nodes [100].

Resource management is addressed by frameworks like Mélange, which treats GPU allocation as a cost-aware bin packing problem, optimizing resource use in AI deployments [66]. Knowledge modeling frameworks such as PEOA improve AI solutions' contextual relevance and accuracy through property graphs and teacher-student transfer learning [101].

Frameworks like NGAI, which process both text and image data, illustrate the importance of multimodal capabilities for compatibility with diverse simulation software [27]. Manticore integrates neural architecture search and mechanistic architecture design to foster robust AI systems [84].

The evolution of these architectural patterns is critical for maximizing LLM potential across applications. Emphasizing modularity and resource efficiency, these frameworks enhance AI system functionality and reliability [2].

6.2 Scalability and Performance Optimization

Scalability and performance optimization are crucial for deploying LLMs in AI-integrated systems. Techniques like CachedAttention significantly reduce the time to first token and inference costs, maintaining high performance while scaling models [7]. The Dynamic Parallel Processing Algorithm (DPPA) combines traditional algorithms with parallel processing to enhance throughput and responsiveness [33].

Experiments reveal diminishing returns in throughput with increased GPU numbers, necessitating optimization strategies for scalability [45]. Future research should develop comprehensive benchmarks encompassing a broader range of devices and performance metrics to enhance understanding of LLM performance in real-world applications [54, 102]. Addressing challenges related to user accents and environmental noise is essential for optimizing speech-enabled systems.

6.3 Security and Robustness in AI Integration

Integrating LLMs into AI architectures requires a comprehensive approach to security and robustness. Developing structured benchmarks for harm detection is critical for creating safer AI systems [16]. LLMs' unpredictability poses challenges for security measures like honeypots, necessitating robust detection and response strategies [35].

Integrating physiological data enhances LLMs' ability to interpret users' psychological states, improving digital psychotherapy tools' effectiveness [1]. Frameworks like OSCAR demonstrate the effectiveness of state machine architectures in enabling dynamic interaction and error recovery [3].

Security is further reinforced by tools like KEN, which ensure the semantic correctness and safety of generated programs [8]. In diplomacy, the Richelieu framework uses past experiences for decision-making, maintaining robustness in complex environments [6]. Ensuring security and robustness

involves structured assessments, adaptive architectures, and rigorous verification processes, enhancing AI systems' resilience [87, 39, 26].

6.4 Emerging Trends and Best Practices

Emerging trends and best practices in AI-integrated system architectures focus on enhancing functionality, efficiency, and ethical considerations. Integrating LLMs into creative processes emphasizes accommodating diverse user needs and improving reflection capabilities [59]. Multimodal interactions, supported by frameworks like OS-Copilot, reflect the trend of integrating AI capabilities across various modalities [97].

Enhancing AI models' adaptability to evolving software environments is critical, with research focusing on using trace data for AIOps [62]. The integration of multi-modal models and refinement of instruction sets are pivotal for enhancing real-time application performance [24].

Transparency and accountability in AI systems are gaining importance, with research focusing on creating adaptable transparency frameworks [11]. Security and privacy remain paramount, with research directed towards robust security alignment measures and privacy-preserving technologies [78].

The democratization of LLM technology in network management through open-source models is expected to drive innovation [103]. In federated learning, improving aggregation techniques and exploring secure computation methods are promising research directions for enhancing scalability and security [56].

Benchmarking efforts provide insights into AI performance and guide the development of more effective systems [104]. These trends and practices ensure AI-integrated systems' functionality, efficiency, and ethical deployment across domains. As AI research progresses, notably with LLMs like ChatGPT, these initiatives are expected to catalyze substantial advancements in AI technologies and applications [38, 71].

7 Natural Language Processing and Human-Machine Interaction

7.1 Advancements in NLP Through LLM Integration

The integration of Large Language Models (LLMs) into Natural Language Processing (NLP) has significantly propelled the field by enhancing the capabilities of NLP systems across diverse applications. Models like ChatGPT have revolutionized text generation, translation, and question-answering, fundamentally reshaping traditional NLP methodologies [38]. These models leverage extensive datasets and sophisticated architectures to produce human-like text, tackling complex linguistic challenges.

LLMs have introduced novel approaches to linguistic modeling, offering insights into language structures that were previously unattainable [105]. This has led to the creation of advanced NLP applications capable of generating text with fluency and coherence, notably improving machine translation accuracy and contextual relevance.

Furthermore, LLMs have enhanced NLP performance in tasks requiring deep semantic understanding, such as sentiment analysis and named entity recognition. Their nuanced comprehension of language has spurred innovations across domains, including systematic literature reviews, legal compliance analysis, and crisis scenario machine translation. For instance, LLMs excel in data extraction for systematic reviews and show promise in automating legal content classification, reducing manual workloads while improving accuracy. Their application in low-resource language translation during crises underscores their versatility in urgent communication needs [30, 106, 10, 12].

The integration of LLMs has markedly improved the efficiency and effectiveness of existing applications, enabling exploration of new linguistic phenomena. For example, models like GPT-4 achieve a mean precision of 83.0

7.2 Enhancing Human-Machine Interaction

The integration of NLP technologies into human-machine interaction systems has markedly improved communication quality and efficiency. Leveraging LLM capabilities, NLP technologies have enhanced machines' abilities to accurately interpret and generate human language, leading to more intuitive interactions and automating tasks such as systematic literature reviews, where LLMs excel in data extraction and processing long texts [106, 10].

A key advancement is the development of systems capable of understanding and responding to natural language inputs conversationally, fostering human-like dialogues and enhancing user experience. For instance, LLMs are employed in customer service applications to generate timely and contextually relevant responses, thereby improving interaction quality [67].

Moreover, NLP technologies enable personalized user experiences by allowing machines to adapt to individual preferences and communication styles through user interaction analysis and machine learning techniques. This personalization is crucial for applications such as virtual assistants and interactive educational tools, resulting in more relevant and engaging interactions [70].

In addition to dialogue systems, NLP technologies have improved information accessibility by efficiently processing and summarizing extensive text volumes. This capability enhances information retrieval systems, allowing users to access succinct summaries of complex documents, expediting the information-seeking process and improving satisfaction. Recent advancements in LLMs, such as OpenAI's ChatGPT, have demonstrated remarkable effectiveness in generating high-quality summaries, facilitating efficient navigation through vast information [102, 9, 31, 10, 106].

The integration of NLP technologies, particularly LLMs, has significantly enhanced communication quality, user engagement, and accessibility. Successful applications in tasks such as text summarization illustrate the superior performance of models like ChatGPT in generating coherent and contextually relevant summaries. However, the responsible deployment of these technologies necessitates a focus on transparency, ensuring that diverse stakeholders can effectively understand and utilize LLM-infused applications. The ongoing development of these systems addresses communication barriers and fosters a more inclusive interaction landscape [31, 11]. As NLP technologies continue to evolve, they hold the potential to further transform human-machine interactions, making them more seamless, intuitive, and effective across various applications and domains.

7.3 Applications in Diverse Contexts

Natural Language Processing (NLP) has significantly transformed user experiences across various industries by leveraging LLM capabilities to enhance interaction quality and operational efficiency. In healthcare, NLP-driven applications have improved diagnostic accuracy and speed, enabling professionals to interpret patient data more effectively. This is particularly evident in the automation of medical documentation and the extraction of critical insights from unstructured clinical notes, streamlining workflows and reducing administrative burdens [5].

In the financial sector, NLP technologies have revolutionized customer service by enabling real-time analysis of interactions and sentiment, allowing institutions to tailor services to individual customer needs and improve satisfaction. NLP-driven chatbots and virtual assistants provide 24/7 support, delivering timely and accurate responses to inquiries, thereby enhancing the user experience [16].

The education sector has benefited from NLP integration, facilitating personalized learning experiences where AI-driven platforms adapt to students' unique learning styles and paces. This personalization is achieved through the analysis of interaction and performance data, enabling customized content delivery and feedback that supports effective learning outcomes [14].

In e-commerce, NLP applications have improved product search and recommendation systems, allowing consumers to find products that meet their preferences with greater ease and accuracy. By analyzing user queries and behavior, NLP technologies generate personalized recommendations that enhance the shopping experience and drive sales [67].

In the entertainment industry, NLP technologies have been utilized to create engaging and interactive content. For example, AI-driven storytelling applications leverage NLP to generate dynamic narratives that adapt to user inputs, providing personalized entertainment experiences that captivate audiences [17].

The application of NLP across diverse contexts has led to significant improvements in user experience. By facilitating more intuitive, efficient, and personalized interactions, NLP technologies are driving innovation and enhancing the quality of services and products available to consumers. This transformation is largely attributed to LLM capabilities, which enhance recommender systems, text summarization, and are being scrutinized for fairness in their applications. Consequently, businesses can leverage these advancements to better understand user preferences, generate high-quality content, and ensure equitable outcomes in critical use cases [107, 31, 69, 71].

8 Conclusion

8.1 Future Directions and Research Opportunities

The integration of Large Language Models (LLMs) into AI systems presents a wealth of research opportunities and potential technological advancements across multiple domains. One critical area of focus is the enhancement of LLM robustness and operational efficiency in optimization tasks, which calls for the development of novel frameworks and a more profound understanding of their foundational processes. In healthcare, the emphasis should be on improving data quality and addressing ethical concerns related to patient data privacy, as the integration of LLMs into clinical workflows could significantly enhance diagnostic precision and treatment planning. Furthermore, extending frameworks to accommodate complex multi-agent scenarios is essential for enhancing LLM adaptability in environments characterized by incomplete information, thereby increasing their effectiveness in dynamic contexts.

In education, the incorporation of LLMs into curricula offers substantial research potential, particularly in tackling ethical issues and assessing the impact of AI on student motivation and learning outcomes. Additionally, optimizing resource management frameworks for dynamic workloads and integrating them with systems that support GPU availability and autoscaling can improve cost-efficiency and performance in AI deployments. Future research should also focus on refining algorithms for larger datasets and exploring their applicability across various fields, highlighting areas ripe for AI integration.

Advancements in cache management strategies and the integration of innovative attention mechanisms with existing LLM architectures could further boost system efficiency. Moreover, enhancing the robustness of systems against complex prompts and exploring additional methods to improve the accuracy of synthesized programs are crucial research areas. Effective methods for iterative improvement in creative applications are vital for deepening our understanding of AI effectiveness.

These research directions and technological advancements hold the promise of significantly improving AI system integration, ensuring that future AI technologies are not only innovative but also responsible and aligned with societal needs and ethical standards.

References

[1] Poorvesh Dongre, Majid Behravan, Kunal Gupta, Mark Billinghurst, and Denis Gračanin. Integrating physiological data with large language models for empathic human-ai interaction, 2024.

[2] James Prather, Paul Denny, Juho Leinonen, Brett A. Becker, Ibrahim Albluwi, Michelle Craig, Hieke Keuning, Natalie Kiesler, Tobias Kohn, Andrew Luxton-Reilly, Stephen MacNeil, Andrew Peterson, Raymond Pettit, Brent N. Reeves, and Jaromir Savelka. The robots are here: Navigating the generative ai revolution in computing education, 2023.

[3] Xiaoqiang Wang and Bang Liu. Oscar: Operating system control via state-aware reasoning and re-planning, 2024.

[4] Sen Huang, Kaixiang Yang, Sheng Qi, and Rui Wang. When large language model meets optimization, 2024.

[5] Hanyao Huang, Ou Zheng, Dongdong Wang, Jiayi Yin, Zijin Wang, Shengxuan Ding, Heng Yin, Chuan Xu, Renjie Yang, Qian Zheng, and Bing Shi. Chatgpt for shaping the future of dentistry: The potential of multi-modal large language model, 2023.

[6] Zhenyu Guan, Xiangyu Kong, Fangwei Zhong, and Yizhou Wang. Richelieu: Self-evolving llm-based agents for ai diplomacy, 2024.

[7] Bin Gao, Zhuomin He, Puru Sharma, Qingxuan Kang, Djordje Jevdjic, Junbo Deng, Xingkun Yang, Zhou Yu, and Pengfei Zuo. Cost-efficient large language model serving for multi-turn conversations with cachedattention, 2024.

[8] Yusheng Zheng, Yiwei Yang, Maolin Chen, and Andrew Quinn. Ken: Kernel extensions using natural language, 2023.

[9] Bumjin Park and Jaesik Choi. Memorizing documents with guidance in large language models, 2024.

[10] Dmitry Scherbakov, Nina Hubig, Vinita Jansari, Alexander Bakumenko, and Leslie A. Lenert. The emergence of large language models (llm) as a tool in literature reviews: an llm automated systematic review, 2024.

[11] Q. Vera Liao and Jennifer Wortman Vaughan. Ai transparency in the age of llms: A human-centered research roadmap, 2023.

[12] Séamus Lankford and Andy Way. Leveraging llms for mt in crisis scenarios: a blueprint for low-resource languages, 2024.

[13] Qin Chen, Jinfeng Ge, Huaqing Xie, Xingcheng Xu, and Yanqing Yang. Large language models at work in china's labor market, 2023.

[14] Yiheng Liu, Tianle Han, Siyuan Ma, Jiayue Zhang, Yuanyuan Yang, Jiaming Tian, Hao He, Antong Li, Mengshen He, Zhengliang Liu, Zihao Wu, Lin Zhao, Dajiang Zhu, Xiang Li, Ning Qiang, Dingang Shen, Tianming Liu, and Bao Ge. Summary of chatgpt-related research and perspective towards the future of large language models, 2023.

[15] Ernests Lavrinovics, Russa Biswas, Johannes Bjerva, and Katja Hose. Knowledge graphs, large language models, and hallucinations: An nlp perspective, 2024.

[16] Swapnaja Achintalwar, Adriana Alvarado Garcia, Ateret Anaby-Tavor, Ioana Baldini, Sara E. Berger, Bishwaranjan Bhattacharjee, Djallel Bouneffouf, Subhajit Chaudhury, Pin-Yu Chen, Lamogha Chiazor, Elizabeth M. Daly, Kirushikesh DB, Rogério Abreu de Paula, Pierre Dognin, Eitan Farchi, Soumya Ghosh, Michael Hind, Raya Horesh, George Kour, Ja Young Lee, Nishtha Madaan, Sameep Mehta, Erik Miehling, Keerthiram Murugesan, Manish Nagireddy, Inkit Padhi, David Piorkowski, Ambrish Rawat, Orna Raz, Prasanna Sattigeri, Hendrik Strobelt, Sarathkrishna Swaminathan, Christoph Tillmann, Aashka Trivedi, Kush R. Varshney, Dennis Wei, Shalisha Witherspooon, and Marcel Zalmanovici. Detectors for safe and reliable llms: Implementations, uses, and limitations, 2024.

[17] Jeremy Straub and Zach Johnson. Initial development and evaluation of the creative artificial intelligence through recurring developments and determinations (cairdd) system, 2024.

[18] Hanguang Xiao, Feizhong Zhou, Xingyue Liu, Tianqi Liu, Zhipeng Li, Xin Liu, and Xiaoxuan Huang. A comprehensive survey of large language models and multimodal large language models in medicine, 2024.

[19] Zhenchang Xing, Qing Huang, Yu Cheng, Liming Zhu, Qinghua Lu, and Xiwei Xu. Prompt sapper: Llm-empowered software engineering infrastructure for ai-native services, 2023.

[20] Kaiyu Huang, Fengran Mo, Xinyu Zhang, Hongliang Li, You Li, Yuanchi Zhang, Weijian Yi, Yulong Mao, Jinchen Liu, Yuzhuang Xu, Jinan Xu, Jian-Yun Nie, and Yang Liu. A survey on large language models with multilingualism: Recent advances and new frontiers, 2025.

[21] Xuechen Liang, Meiling Tao, Yinghui Xia, Tianyu Shi, Jun Wang, and JingSong Yang. Cmat: A multi-agent collaboration tuning framework for enhancing small language models, 2024.

[22] Yuan Yang, Siheng Xiong, Ehsan Shareghi, and Faramarz Fekri. The compressor-retriever architecture for language model os, 2024.

[23] Isamu Isozaki, Manil Shrestha, Rick Console, and Edward Kim. Towards automated penetration testing: Introducing llm benchmark, analysis, and improvements, 2025.

[24] Antonio A. Ginart, Naveen Kodali, Jason Lee, Caiming Xiong, Silvio Savarese, and John Emmons. Asynchronous tool usage for real-time agents, 2024.

[25] Can Cui, Yunsheng Ma, Xu Cao, Wenqian Ye, and Ziran Wang. Drive as you speak: Enabling human-like interaction with large language models in autonomous vehicles, 2023.

[26] Yufan Chen, Arjun Arunasalam, and Z. Berkay Celik. Can large language models provide security privacy advice? measuring the ability of llms to refute misconceptions, 2023.

[27] Jiajing Chen, Weihang Xu, Haiming Cao, Zihuan Xu, Yu Zhang, Zhao Zhang, and Siyao Zhang. Multimodal road network generation based on large language model, 2024.

[28] Venkat Srinivasan, Darshan Gandhi, Urmish Thakker, and Raghu Prabhakar. Training large language models efficiently with sparsity and dataflow, 2023.

[29] Bosheng Ding, Chengwei Qin, Ruochen Zhao, Tianze Luo, Xinze Li, Guizhen Chen, Wenhan Xia, Junjie Hu, Anh Tuan Luu, and Shafiq Joty. Data augmentation using large language models: Data perspectives, learning paradigms and challenges, 2024.

[30] Shabnam Hassani. Enhancing legal compliance and regulation analysis with large language models, 2024.

[31] Lochan Basyal and Mihir Sanghvi. Text summarization using large language models: A comparative study of mpt-7b-instruct, falcon-7b-instruct, and openai chat-gpt models, 2023.

[32] Ellen Jiang, Kristen Olson, Edwin Toh, Alejandra Molina, Aaron Donsbach, Michael Terry, and Carrie J Cai. Promptmaker: Prompt-based prototyping with large language models. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*, pages 1–8, 2022.

[33] Antonello Ceravola, Frank Joublin, Ahmed R. Sadik, Bram Bolder, and Juha-Pekka Tolvanen. Hypergraphos: A meta operating system for science and engineering, 2024.

[34] Yuanqing Yu, Zhefan Wang, Weizhi Ma, Shuai Wang, Chuhan Wu, Zhiqiang Guo, and Min Zhang. Steptool: Enhancing multi-step tool usage in llms through step-grained reinforcement learning, 2025.

[35] Hakan T. Otal and M. Abdullah Canbaz. Llm honeypot: Leveraging large language models as advanced interactive honeypot systems, 2024.

[36] Javier García Gilabert, Carlos Escolano, Aleix Sant Savall, Francesca De Luca Fornaciari, Audrey Mash, Xixian Liao, and Maite Melero. Investigating the translation capabilities of large language models trained on parallel data only, 2024.

[37] Bin Hong, Jinze Wu, Jiayu Liu, Liang Ding, Jing Sha, Kai Zhang, Shijin Wang, and Zhenya Huang. End-to-end graph flattening method for large language models, 2024.

[38] Frank Joublin, Antonello Ceravola, Joerg Deigmoeller, Michael Gienger, Mathias Franzius, and Julian Eggert. A glimpse in chatgpt capabilities and its impact for ai research, 2023.

[39] Mike Perkins. Academic integrity considerations of ai large language models in the post-pandemic era: Chatgpt and beyond. *Journal of University Teaching and Learning Practice*, 20(2), 2023.

[40] Vahid Mohammadi Safarzadeh and Hamed Ghasr Loghmani. Artificial intelligence in the low-level realm – a survey, 2021.

[41] Gabriele Tolomei, Cesare Campagnano, Fabrizio Silvestri, and Giovanni Trappolini. Prompt-to-os (p2os): Revolutionizing operating systems and human-computer interaction with integrated ai generative models, 2023.

[42] Yingqiang Ge, Yujie Ren, Wenyue Hua, Shuyuan Xu, Juntao Tan, and Yongfeng Zhang. Llm as os, agents as apps: Envisioning aios, agents and the aios-agent ecosystem, 2023.

[43] Aditya K Kamath and Sujay Yadalam. Herding llamas: Using llms as an os module, 2024.

[44] Kai Mei, Zelong Li, Shuyuan Xu, Ruosong Ye, Yingqiang Ge, and Yongfeng Zhang. Aios: Llm agent operating system, 2024.

[45] Jared Fernandez, Luca Wehrstedt, Leonid Shamis, Mostafa Elhoushi, Kalyan Saladi, Yonatan Bisk, Emma Strubell, and Jacob Kahn. Hardware scaling trends and diminishing returns in large-scale distributed training, 2024.

[46] Justin Del Vecchio, Andrew Perreault, and Eliana Furmanek. Dynamic code orchestration: Harnessing the power of large language models for adaptive script execution, 2024.

[47] Musfiqur Rahman, SayedHassan Khatoonabadi, Ahmad Abdellatif, Haya Samaana, and Emad Shihab. On the variability of ai-based software systems due to environment configurations, 2024.

[48] Abi Aryan, Aakash Kumar Nain, Andrew McMahon, Lucas Augusto Meyer, and Harpreet Singh Sahota. The costly dilemma: Generalization, evaluation and cost-optimal deployment of large language models, 2023.

[49] Kamlesh Sharma, T. Suryakanthi, and T. V. Prasad. Exploration of speech enabled system for english, 2013.

[50] Jeremy Kepner, Ron Brightwell, Alan Edelman, Vijay Gadepally, Hayden Jananthan, Michael Jones, Sam Madden, Peter Michaleas, Hamed Okhravi, Kevin Pedretti, Albert Reuther, Thomas Sterling, and Mike Stonebraker. Tabularosa: Tabular operating system architecture for massively parallel heterogeneous compute engines, 2018.

[51] Johan Kristiansson. Colonyos – a meta-operating system for distributed computing across heterogeneous platform, 2024.

[52] Rithesh Murthy, Liangwei Yang, Juntao Tan, Tulika Manoj Awalgaonkar, Yilun Zhou, Shelby Heinecke, Sachin Desai, Jason Wu, Ran Xu, Sarah Tan, Jianguo Zhang, Zhiwei Liu, Shirley Kokane, Zuxin Liu, Ming Zhu, Huan Wang, Caiming Xiong, and Silvio Savarese. Mobileaibench: Benchmarking llms and lmms for on-device use cases, 2024.

[53] Achilleas Santi Seisa, Sumeet Gajanan Satpute, and George Nikolakopoulos. Comparison between docker and kubernetes based edge architectures for enabling remote model predictive control for aerial robots, 2022.

[54] Jie Xiao, Qianyi Huang, Xu Chen, and Chen Tian. Large language model performance benchmarking on mobile platforms: A thorough evaluation, 2024.

[55] Herbert Woisetschläger, Alexander Isenko, Shiqiang Wang, Ruben Mayer, and Hans-Arno Jacobsen. Federated fine-tuning of llms on the very edge: The good, the bad, the ugly, 2024.

[56] Daniel Madrigal Diaz, Andre Manoel, Jialei Chen, Nalin Singal, and Robert Sim. Project florida: Federated learning made easy, 2023.

[57] Xinyi Hou, Yanjie Zhao, Yue Liu, Zhou Yang, Kailong Wang, Li Li, Xiapu Luo, David Lo, John Grundy, and Haoyu Wang. Large language models for software engineering: A systematic literature review. *ACM Transactions on Software Engineering and Methodology*, 33(8):1–79, 2024.

[58] Weize Liu, Guocong Li, Kai Zhang, Bang Du, Qiyuan Chen, Xuming Hu, Hongxia Xu, Jintai Chen, and Jian Wu. Mind's mirror: Distilling self-evaluation capability and comprehensive thinking from large language models, 2024.

[59] Anqi Wang, Zhizhuo Yin, Yulu Hu, Yuanyuan Mao, and Pan Hui. Exploring the potential of large language models in artistic creation: Collaboration and reflection on creative programming, 2024.

[60] Bin Lin, Chen Zhang, Tao Peng, Hanyu Zhao, Wencong Xiao, Minmin Sun, Anmin Liu, Zhipeng Zhang, Lanbo Li, Xiafei Qiu, Shen Li, Zhigang Ji, Tao Xie, Yong Li, and Wei Lin. Infinite-llm: Efficient llm service for long context with distattention and distributed kvcache, 2024.

[61] Deepak Narayanan, Mohammad Shoeybi, Jared Casper, Patrick LeGresley, Mostofa Patwary, Vijay Anand Korthikanti, Dmitri Vainbrand, Prethvi Kashinkunti, Julie Bernauer, Bryan Catanzaro, Amar Phanishayee, and Matei Zaharia. Efficient large-scale language model training on gpu clusters using megatron-lm, 2021.

[62] Lingzhe Zhang, Tong Jia, Mengxi Jia, Yifan Wu, Aiwei Liu, Yong Yang, Zhonghai Wu, Xuming Hu, Philip S. Yu, and Ying Li. A survey of aiops for failure management in the era of large language models, 2024.

[63] Karim Benharrak, Tim Zindulka, and Daniel Buschek. Deceptive patterns of intelligent and interactive writing assistants, 2024.

[64] Youngsuk Park, Kailash Budhathoki, Liangfu Chen, Jonas Kübler, Jiaji Huang, Matthäus Kleindessner, Jun Huan, Volkan Cevher, Yida Wang, and George Karypis. Inference optimization of foundation models on ai accelerators, 2024.

[65] Hongcheng Guo, Jian Yang, Jiaheng Liu, Liqun Yang, Linzheng Chai, Jiaqi Bai, Junran Peng, Xiaorong Hu, Chao Chen, Dongfeng Zhang, Xu Shi, Tieqiao Zheng, Liangfan Zheng, Bo Zhang, Ke Xu, and Zhoujun Li. Owl: A large language model for it operations, 2024.

[66] Tyler Griggs, Xiaoxuan Liu, Jiaxiang Yu, Doyoung Kim, Wei-Lin Chiang, Alvin Cheung, and Ion Stoica. Mélange: Cost efficient large language model serving by exploiting gpu heterogeneity, 2024.

[67] Mazal Bethany, Athanasios Galiopoulos, Emet Bethany, Mohammad Bahrami Karkevandi, Nishant Vishwamitra, and Peyman Najafirad. Large language model lateral spear phishing: A comparative study in large-scale organizational settings, 2024.

[68] Ali Borji. A categorical archive of chatgpt failures, 2023.

[69] Vincent Freiberger and Erik Buchmann. Fairness certification for natural language processing and large language models, 2024.

[70] Yuqi Yang, Xiaowen Huang, and Jitao Sang. Exploring the privacy protection capabilities of chinese large language models, 2024.

[71] Ximing Lu, Melanie Sclar, Skyler Hallinan, Niloofar Mireshghallah, Jiacheng Liu, Seungju Han, Allyson Ettinger, Liwei Jiang, Khyathi Chandu, Nouha Dziri, and Yejin Choi. Ai as humanity's salieri: Quantifying linguistic creativity of language models via systematic attribution of machine text against web text, 2025.

[72] Xinyi Mou, Jingcong Liang, Jiayu Lin, Xinnong Zhang, Xiawei Liu, Shiyue Yang, Rong Ye, Lei Chen, Haoyu Kuang, Xuanjing Huang, and Zhongyu Wei. Agentsense: Benchmarking social intelligence of language agents through interactive scenarios, 2024.

[73] Jianian Gong, Nachuan Duan, Ziheng Tao, Zhaohui Gong, Yuan Yuan, and Minlie Huang. How well do large language models serve as end-to-end secure code producers?, 2024.

[74] Kongyang Chen, Zixin Wang, Bing Mi, Waixi Liu, Shaowei Wang, Xiaojun Ren, and Jiaxing Shen. Machine unlearning in large language models, 2024.

[75] Hongsun Jang, Jaeyong Song, Jaewon Jung, Jaeyoung Park, Youngsok Kim, and Jinho Lee. Smart-infinity: Fast large language model training using near-storage processing on a real system, 2024.

[76] N'yoma Diamond. Ai does not alter perceptions of text messages, 2024.

[77] Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. Encouraging divergent thinking in large language models through multi-agent debate, 2024.

[78] Yihe Fan, Yuxin Cao, Ziyu Zhao, Ziyao Liu, and Shaofeng Li. Unbridled icarus: A survey of the potential perils of image inputs in multimodal large language model security, 2024.

[79] Guodong Du, Jing Li, Hanting Liu, Runhua Jiang, Shuyang Yu, Yifei Guo, Sim Kuan Goh, and Ho-Kin Tang. Knowledge fusion by evolving weights of language models, 2024.

[80] Soyeon Caren Han, Feiqi Cao, Josiah Poon, and Roberto Navigli. Multimodal large language models and tunings: Vision, language, sensors, audio, and beyond, 2024.

[81] Ziqiang Zheng, Jipeng Zhang, Tuan-Anh Vu, Shizhe Diao, Yue Him Wong Tim, and Sai-Kit Yeung. Marinegpt: Unlocking secrets of ocean to the public, 2023.

[82] Yizhang Jin, Jian Li, Yexin Liu, Tianjun Gu, Kai Wu, Zhengkai Jiang, Muyang He, Bo Zhao, Xin Tan, Zhenye Gan, Yabiao Wang, Chengjie Wang, and Lizhuang Ma. Efficient multimodal large language models: A survey, 2024.

[83] Jumbly Grindrod. Large language models and linguistic intentionality, 2024.

[84] Nicholas Roberts, Samuel Guo, Zhiqi Gao, Satya Sai Srinath Namburi GNVV, Sonia Cromp, Chengjun Wu, Chengyu Duan, and Frederic Sala. Pretrained hybrids with mad skills, 2024.

[85] Salma Afifi, Febin Sunny, Mahdi Nikdast, and Sudeep Pasricha. Accelerating neural networks for large language models and graph processing with silicon photonics, 2024.

[86] Yeonhong Park, Jake Hyun, SangLyul Cho, Bonggeun Sim, and Jae W. Lee. Any-precision llm: Low-cost deployment of multiple, different-sized llms, 2024.

[87] Suriya Prakash Jambunathan, Ashwath Shankarnarayan, and Parijat Dube. Convnlp: Image-based ai text detection, 2024.

[88] Sorin Grigorescu and Mihai Zaha. Cybercortex.ai: An ai-based operating system for autonomous robotics and complex automation, 2024.

[89] Antonello Ceravola and Frank Joublin. Hypergraphos: A modern meta-operating system for the scientific and engineering domains, 2024.

[90] Ken Gu, Ruoxi Shang, Tim Althoff, Chenglong Wang, and Steven M. Drucker. How do analysts understand and verify ai-assisted data analyses?, 2024.

[91] Gerd Kortemeyer. Tailoring chatbots for higher education: Some insights and experiences, 2024.

[92] Stephen MacNeil, Andrew Tran, Joanne Kim, Ziheng Huang, Seth Bernstein, and Dan Mogil. Prompt middleware: Mapping prompts for large language models to ui affordances, 2023.

[93] Hao Zhou, Chengming Hu, Ye Yuan, Yufei Cui, Yili Jin, Can Chen, Haolun Wu, Dun Yuan, Li Jiang, Di Wu, Xue Liu, Charlie Zhang, Xianbin Wang, and Jiangchuan Liu. Large language model (llm) for telecommunications: A comprehensive survey on principles, key techniques, and opportunities, 2024.

[94] Zhiwei Liu, Weiran Yao, Jianguo Zhang, Le Xue, Shelby Heinecke, Rithesh Murthy, Yihao Feng, Zeyuan Chen, Juan Carlos Niebles, Devansh Arpit, Ran Xu, Phil Mui, Huan Wang, Caiming Xiong, and Silvio Savarese. Bolaa: Benchmarking and orchestrating llm-augmented autonomous agents, 2023.

[95] Stephen Casper, Jason Lin, Joe Kwon, Gatlen Culp, and Dylan Hadfield-Menell. Explore, establish, exploit: Red teaming language models from scratch, 2023.

[96] Hongwei Jin, George Papadimitriou, Krishnan Raghavan, Pawel Zuk, Prasanna Balaprakash, Cong Wang, Anirban Mandal, and Ewa Deelman. Large language models for anomaly detection in computational workflows: from supervised fine-tuning to in-context learning, 2024.

[97] Zhiyong Wu, Chengcheng Han, Zichen Ding, Zhenmin Weng, Zhoumianze Liu, Shunyu Yao, Tao Yu, and Lingpeng Kong. Os-copilot: Towards generalist computer agents with self-improvement, 2024.

[98] Donghoon Shin, Gary Hsieh, and Young-Ho Kim. Planfitting: Tailoring personalized exercise plans with large language models, 2023.

[99] Zelong Li, Shuyuan Xu, Kai Mei, Wenyue Hua, Balaji Rama, Om Raheja, Hao Wang, He Zhu, and Yongfeng Zhang. Autoflow: Automated workflow generation for large language model agents, 2024.

[100] Ye Li. Graph-of-thought: Utilizing large language models to solve complex and dynamic business problems, 2024.

[101] Sakhinana Sagar Srinivas, Vijay Sri Vaikunth, and Venkataramana Runkana. Knowledge graph modeling-driven large language model operating system (llm os) for task automation in process engineering problem-solving, 2024.

[102] Qusai Khraisha, Sophie Put, Johanna Kappenberg, Azza Warraitch, and Kristin Hadfield. Can large language models replace humans in the systematic review process? evaluating gpt-4's efficacy in screening and extracting data from peer-reviewed and grey literature in multiple languages, 2023.

[103] Dimitrios Michael Manias, Ali Chouman, and Abdallah Shami. Semantic routing for enhanced performance of llm-assisted intent-based 5g core network management and orchestration, 2024.

[104] Mingyu Jin, Qinkai Yu, Dong Shu, Haiyan Zhao, Wenyue Hua, Yanda Meng, Yongfeng Zhang, and Mengnan Du. The impact of reasoning step length on large language models, 2024.

[105] Jumbly Grindrod. Modelling language, 2024.

[106] Yutong Hu, Quzhe Huang, Mingxu Tao, Chen Zhang, and Yansong Feng. Can perplexity reflect large language model's ability in long text understanding?, 2024.

[107] Zihuai Zhao, Wenqi Fan, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Zhen Wen, Fei Wang, Xiangyu Zhao, Jiliang Tang, and Qing Li. Recommender systems in the era of large language models (llms), 2024.

Disclaimer:

SurveyX is an AI-powered system designed to automate the generation of surveys. While it aims to produce high-quality, coherent, and comprehensive surveys with accurate citations, the final output is derived from the AI's synthesis of pre-processed materials, which may contain limitations or inaccuracies. As such, the generated content should not be used for academic publication or formal submissions and must be independently reviewed and verified. The developers of SurveyX do not assume responsibility for any errors or consequences arising from the use of the generated surveys.