Large Language Models for Task Automation: A Comprehensive Review of Techniques and Implementation Challenges

Jaehyun Chung, Minji Lee, and Joongheon Kim
Department of Electrical and Computer Engineering, Korea University, Seoul, Republic of Korea
E-mails: rupang1234@korea.ac.kr, vminji@naver.com, joongheon@korea.ac.kr

***Abstract*—Large language models (LLMs) have rapidly advanced task automation by enabling high-quality natural language understanding, reasoning, and multi-modal interaction across domains such as software development, healthcare, customer service, and scientific discovery. Despite their transformative potential, challenges such as high computational cost, domain adaptation, safety, and robust error handling remain significant barriers to practical deployment. Recent research has addressed these issues through techniques including parameter-efficient fine-tuning, retrieval-augmented generation, domain-specific pretraining, tool integration, and multi-agent orchestration frameworks, which collectively improve scalability, adaptability, and reliability. This paper presents a comprehensive survey of state-of-the-art approaches to LLM-driven task automation, systematically categorizing methods by optimization strategy, integration architecture, and application scenario. Implementation challenges are further analyzed, and emerging solutions are reviewed that aim to balance performance, efficiency, and trustworthiness. This review provides a synthesized perspective on how LLMs are evolving toward robust, context-aware automation systems suitable for real-world deployment.**

***Index Terms*—Large Language Models, Task Automation, Multi-Agent Systems, AI Integration**

I. Introduction

In recent years, artificial intelligence has experienced remarkable progress across multiple domains, including natural language processing, computer vision, and decision-making systems [1]. Among these advancements, large language models (LLMs) have emerged as a core enabler of next-generation automation systems [2]. Leveraging extensive pretraining on massive text and multi-modal datasets, LLMs have demonstrated exceptional capabilities in natural language understanding, reasoning, and content generation [3]. These strengths have facilitated their integration into diverse application scenarios, such as software engineering, healthcare diagnostics, legal document analysis, customer support, and scientific discovery. One of the most powerful aspects of LLMs is their ability to generalize across tasks through few-shot and zero-shot learning, enabling rapid adaptation without extensive domain-specific retraining [4], [5]. Combined with emerging frameworks for tool augmentation and application programming interface (API) orchestration, LLMs are increasingly positioned as central agents in automated workflows that require complex reasoning and multi-step decision-making.

However, the impressive capabilities of LLM-based automation systems come with significant challenges. High computational costs and large memory requirements pose barriers to deployment in resource-constrained environments [6]. At the same time, prompt sensitivity, hallucination errors, domain adaptation limitations, latency in real-time systems, and safety concerns create additional hurdles for robust and trustworthy adoption [7], [8]. Deploying an LLM-based automation agent in mission-critical settings such as medical diagnosis or legal reasoning requires not only high accuracy but also reliable fail-safe mechanisms and verifiable outputs. To address these challenges, recent research has explored a range of strategies to improve efficiency, adaptability, and reliability, including parameter-efficient fine-tuning, retrieval-augmented generation, domain-specific pretraining, multi-agent collaboration, and tool integration for external system interaction. Such approaches have shown promise in mitigating current limitations while expanding the range of feasible deployment scenarios. This paper presents a comprehensive review of recent techniques and frameworks for LLM-driven task automation. The analysis covers optimization strategies, integration architectures, and representative application cases, and also examines implementation challenges alongside emerging solutions aimed at enhancing efficiency, scalability, and reliability in diverse operational environments.

This paper is organized as follows. Section II outlines the architecture and key principles of large language models. Section III reviews recent techniques and frameworks for LLM-driven task automation. Section IV concludes the paper and suggests future research directions.

II. Architecture of Large Language Models

LLMs are built upon the Transformer architecture, which, as illustrated in Fig. 1, relies on self-attention mechanisms to capture long-range dependencies and represent contextual relationships between tokens in a scalable manner [9]. The attention mechanism allows each position to attend to every other, integrating information from the entire sequence [10]. While the Transformer framework provides a unifying foundation, LLMs

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2024-RS-2024-00436887) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation); and also by IITP grant funded by MSIT (RS-2024-00439803, SW Star Lab). *(Corresponding author: Joongheon Kim)*

can be implemented in three main configurations: encoder-only, encoder–decoder, and decoder-only, each optimized for different categories of tasks.

Encoder-only architectures process sequences bidirectionally, enabling each token to attend to both preceding and succeeding tokens during representation learning. This design produces rich semantic representations, which are particularly effective for understanding-focused tasks such as classification, ranking, or retrieval. The absence of masking in the attention mechanism allows full visibility across the sequence, capturing complex contextual relationships. Encoder–decoder architectures consist of two components: the encoder converts the input into latent representations, and the decoder generates the output sequence using both self-attention and cross-attention. This structure is well-suited for tasks where input and output differ in length or modality, including translation, summarization, and structured data generation. Decoder-only architectures stack multiple Transformer decoder layers, each containing masked multi-head self-attention to ensure predictions depend only on preceding tokens. A position-wise feed-forward network follows to expand representational capacity, with residual connections and normalization layers to stabilize training. Given an input $X \in \mathbb{R}^{n \times d}$, self-attention can be expressed as,

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V, \tag{1}$$

where Q, K, and V are the query, key, and value matrices from learned projections of X, and d_k is the key vector dimension.

Across all configurations, positional encodings are added to token embeddings to represent sequence order, compensating for the attention mechanism's lack of inherent positional awareness. These encodings may be fixed sinusoidal patterns or learned parameters. Tokenization methods, such as byte pair encoding or sentencepiece, segment text into subword units, balancing vocabulary size, coverage, and efficiency. LLMs are typically pretrained on large-scale corpora with objectives that align with their architectural design. For encoder-only models, masked token prediction is used to reconstruct hidden tokens from context. Encoder–decoder models are trained with sequence-to-sequence objectives, optimizing the probability of generating the correct output given input. Decoder-only models use an autoregressive objective, predicting each token from all preceding tokens. After pretraining, models are adapted to downstream tasks through fine-tuning, instruction tuning, or parameter-efficient adaptation methods that update only part of the parameters while retaining the benefits of large-scale pretraining.

III. Techniques and Frameworks for LLM-Driven Task Automation

Recent advancements in LLMs have enabled a broad spectrum of task automation solutions across diverse domains, ranging from administrative workflows and mobile applications to robotics and intelligent transportation. These frameworks combine the language understanding and reasoning capabilities of LLMs with domain-specific tools, APIs, and multi-modal

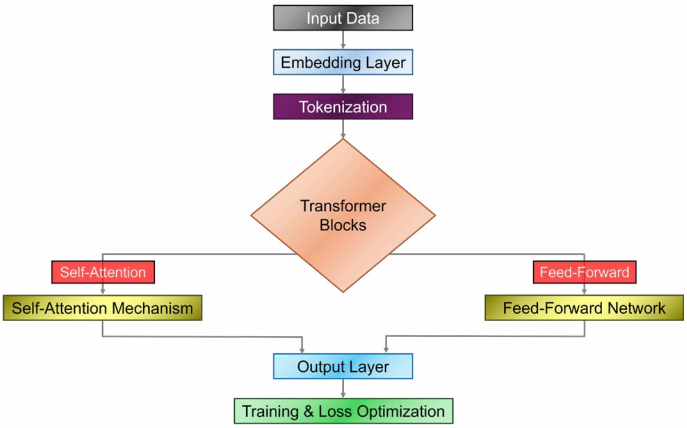

Fig. 1: Overall architecture of the LLM, illustrating the data flow from input representation to output generation via attention-based transformer blocks.

processing to achieve efficient, scalable, and context-aware automation. They often leverage a combination of retrieval-based methods, tool orchestration, and adaptive prompting strategies to align LLM outputs with real-world operational constraints. This section reviews representative approaches, categorized by their target environment and core technical strategies.

One study proposed a healthcare administrative task automation framework to streamline workflows such as patient scheduling, insurance claim processing, and clinical documentation. The system integrates retrieval-augmented generation for accurate domain-specific responses and task planning modules that ensure compliance with healthcare regulations, demonstrating notable time savings for administrative staff while maintaining high data accuracy [11]. Another paper, VisionTasker, introduced a multi-modal mobile automation framework that combines on-device screen capture analysis with LLM-based task planning to interpret UI layouts and icons from visual input, map them to functional actions, and execute them via platform-specific APIs [12]. This design enables automation in applications without direct API access, making it suitable for third-party apps in mobile ecosystems. Complementarily, AutoDroid employed a more API-centric approach, where LLMs generate executable Android Debug Bridge (ADB) commands directly from natural language instructions, prioritizing efficiency in direct command execution [13].

In the area of benchmarking, TaskBench provides a standardized evaluation protocol to measure task success rates, reasoning accuracy, and execution efficiency across diverse automation scenarios, ranging from file management to web-based workflows [14]. The benchmark includes both synthetic and real-world task sets, enabling systematic comparisons across different LLM architectures and prompting techniques. By offering quantitative metrics and reproducible test cases, TaskBench facilitates fair performance evaluation and identifies strengths and weaknesses of competing automation frameworks. In the education domain, another study developed LLM agents for automating tasks such as personalized feedback

generation, grading, and content creation for learners [15]. These agents integrate with learning management systems and adapt their output to pedagogical requirements, employing multi-agent orchestration to decompose complex requests into sub-tasks for reliable and aligned responses. Pilot deployments in higher education settings reported increased grading efficiency, improved consistency in feedback, and higher student satisfaction compared to manual grading. For physical-world automation, the linear programming-integrated large language model (LiP-LLM) combines LLM-based task decomposition with linear programming and dependency graph modeling to coordinate multi-robot task planning [16]. This approach integrates symbolic optimization with natural language reasoning to achieve efficient scheduling and execution in multi-agent robotic environments. In experiments with warehouse robots, LiP-LLM demonstrated significant reductions in idle time and improved overall throughput. In a related direction, VistaGPT targets intelligent transportation systems by applying a generative parallel transformer architecture to process multi-modal traffic data and coordinate vehicle operations [17]. Its ability to perform real-time route planning, traffic flow optimization, and inter-vehicle communication makes it suitable for large-scale deployment in connected autonomous vehicle networks.

Overall, these studies illustrate the breadth of LLM-driven task automation, across domains from digital workflows to physical-world operations. By integrating natural language understanding with vision-based perception, structured planning, and domain-specific tool usage, these frameworks demonstrate how LLMs can be tailored to meet diverse needs while maintaining scalability and reliability across environments.

IV. Conclusion

This paper presented a comprehensive review of recent techniques and frameworks for LLM-driven task automation, covering domains from healthcare and education to robotics and intelligent transportation. By surveying representative research, this paper identifies the core strategies that enable LLMs to act as robust, context-aware agents capable of performing complex workflows with minimal human intervention. The review organized existing work according to architectural design choices, integration approaches, and domain-specific adaptations, providing a structured perspective on the current state of the field. Representative works showcased retrieval-augmented generation for domain accuracy, multi-modal perception for graphical user interface understanding, symbolic optimization for multi-agent coordination, and benchmarking frameworks for standardized evaluation. These approaches demonstrate how LLMs can integrate language reasoning with vision, structured planning, and API-based control to deliver scalable automation solutions. The surveyed studies indicate that LLM-driven automation is moving from experimental prototypes to practical systems for both digital and physical environments. Future research is expected to explore hybrid architectures that merge multi-modal inputs with symbolic reasoning, as well as adaptive execution strategies that adjust computation based on task complexity and constraints. Strengthening safety, interpretability, and resource efficiency will remain key priorities in deploying trustworthy, high-performance automation agents at scale.

V. Acknowledgement

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2024-RS-2024-00436887) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation); and also by IITP grant funded by MSIT (RS-2024-00439803, SW Star Lab). *(Corresponding author: Joongheon Kim)*

References

[1] S. S. Sengar, A. B. Hasan, S. Kumar, and F. Carroll, "Generative artificial intelligence: A systematic review and applications," *Multimedia Tools and Applications*, vol. 84, pp. 23 661–23 700, August 2025.

[2] T. Nie, J. Sun, and W. Ma, "Exploring the roles of large language models in reshaping transportation systems: A survey, framework, and roadmap," *Artificial Intelligence for Transportation*, vol. 1, pp. 100 003–100 033, July 2025.

[3] J. Wu, W. Gan, Z. Chen, S. Wan, and P. S. Yu, "Multimodal large language models: A survey," in *IEEE International Conference on Big Data (BigData)*, Sorrento, Italy, December 2023, pp. 2247–2256.

[4] Y. Dang, H. Li, B. Liu, and X. Zhang, "Cross-domain few-shot learning for hyperspectral image classification based on global-to-local enhanced channel attention," *IEEE Geoscience and Remote Sensing Letters*, vol. 22, pp. 1–5, January 2025.

[5] S. Rahman, S. Khan, and F. Porikli, "A unified approach for conventional zero-shot, generalized zero-shot, and few-shot learning," *IEEE Transactions on Image Processing*, vol. 27, no. 11, pp. 5652–5667, November 2018.

[6] Z. Yao, Y. Xu, H. Xu, Y. Liao, and Z. Xie, "Efficient deployment of large language models on resource-constrained devices," *CoRR*, vol. abs/2501.02438, January 2025.

[7] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin *et al.*, "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," *ACM Transactions on Information Systems*, vol. 43, no. 2, pp. 1–55, January 2025.

[8] E. J. Husom, A. Goknil, M. Astekin, L. K. Shar, A. Kåsen, S. Sen, B. A. Mithassel, and A. Soylu, "Sustainable LLM inference for edge AI: evaluating quantized llms for energy efficiency, output accuracy, and inference latency," *CoRR*, vol. abs/2504.03360, April 2025.

[9] M. A. K. Raiaan, M. S. H. Mukta, K. Fatema, N. M. Fahad, S. Sakib, M. M. J. Mim, J. Ahmad, M. E. Ali, and S. Azam, "A review on large language models: Architectures, applications, taxonomies, open issues and challenges," *IEEE Access*, vol. 12, pp. 26 839–26 874, February 2024.

[10] E. Kasneci, K. Seßler, S. Küchemann, M. Bannert, D. Dementieva, F. Fischer, U. Gasser, G. Groh, S. Günnemann, E. Hüllermeier *et al.*, "ChatGPT for good? On opportunities and challenges of large language models for education," *Learning and individual differences*, vol. 103, pp. 102 274–102 282, April 2023.

[11] S. A. Gebreab, K. Salah, R. Jayaraman, M. Habib ur Rehman, and S. Ellaham, "LLM-based framework for administrative task automation in healthcare," in *Proc. International Symposium on Digital Forensics and Security (ISDFS)*, San Antonio, TX, USA, April 2024, pp. 1–7.

[12] Y. Song, Y. Bian, Y. Tang, G. Ma, and Z. Cai, "VisionTasker: Mobile task automation using vision based UI understanding and LLM task planning," in *Proc. ACM Symposium on User Interface Software and Technology*, New York, NY, USA, October 2024, pp. 1–17.

[13] H. Wen, Y. Li, G. Liu, S. Zhao, T. Yu, T. J.-J. Li, S. Jiang, Y. Liu, Y. Zhang, and Y. Liu, "AutoDroid: LLM-powered task automation in Android," in *Proc. International Conference on Mobile Computing and Networking*, New York, NY, USA, May 2024, pp. 543–557.

[14] Y. Shen, K. Song, X. Tan, W. Zhang, K. Ren, S. Yuan, W. Lu, D. Li, and Y. Zhuang, "TaskBench: Benchmarking large language models for task automation," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 37, Vancouver, BC, Canada, December 2024, pp. 4540–4574.

[15] Z. Chu, S. Wang, J. Xie, T. Zhu, Y. Yan, J. Ye, A. Zhong, X. Hu, J. Liang, P. S. Yu, and Q. Wen, "LLM agents for education: Advances and applications," *CoRR*, vol. abs/2503.11733, March 2025.

[16] K. Obata, T. Aoki, T. Horii, T. Taniguchi, and T. Nagai, "LiP-LLM: Integrating linear programming and dependency graph with large language models for multi-robot task planning," *IEEE Robotics and Automation Letters*, vol. 10, no. 2, pp. 1122–1129, February 2025.

[17] Y. Tian, X. Li, H. Zhang, C. Zhao, B. Li, X. Wang, X. Wang, and F.-Y. Wang, "VistaGPT: Generative parallel transformers for vehicles with intelligent systems for transport automation," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 9, pp. 4198–4207, September 2023.