From Prompts to API Calls: A use Case-Driven Evaluation of Tool Calling in Large Language Models for Email Workflow Automation

Dr. C. Ashwini; Navya Jain;
Kaustuk Saraf; S Sai Rajiv
Department of Computer Science and Engineering
SRM Institute of Science and Technology, Ramapuram
Chennai, India

Abstract — This paper emphasizes an exploration based on surveys on how the different large language models (LLM) respond and perform when they are integrated with the API to automate email administration. The study investigates the ability of each model to understand the indications of natural language, generate consistent responses and execute tasks such as composing, sending and reading emails. Through the tool calling capabilities, these models can directly invoke API functions, allowing the execution of perfect tasks through natural language commands. The call of tools serves as a critical mechanism that allows the LLM to go beyond static responses and interact dynamically with external systems, effectively transforming them into active agents capable of completing user -intended tasks. This paper explores how each model interprets the signatures of the function, select the appropriate tools and handle the interactions of several steps using these capabilities. With this exploration, the article highlights the differences in the capacity of response of the model, precision and contextual understanding in cases of use of the real world. Provides information on the strengths and limitations of each LLM in real world applications. This study additionally analyzes how LLM's behavior varies between models in terms of reliability, latency and context retention.

Keywords—Large Language Models (LLMs), Tool Calling, Email Automation, Natural Language Processing (NLP), Intelligent Email Management, Workflow Automation, Artificial Intelligence (AI).

I.Introduction

Large language models (LLM) have quickly advanced the landscape of artificial intelligence, which demonstrates notable capabilities in the understanding and generation of human text. With the introduction of functions calls, also known as

tool calls, LLMs are no longer limited to passive language generation. On the other hand, they can now actively interact with external tools and services invoking defined functions in structured formats, allowing a more agent and action-oriented operation mode. This change marks a fundamental evolution in how LLMs can be applied to real world's tasks, closing the gap between language understanding and functional execution.

A particularly valuable use case lies in the automation of email workflows, a domain that remains fundamental for professional and professional digital communication. Correos management (composition, response, organization or programming) can take a long time and demand cognitively. The automation of these tasks through LLM that can call the tool not only saves time but also improves general productivity.

With API such as Gmail, it is possible that LLM receives natural language instructions and translates them into

DOI: https://doi.org/10.5281/zenodo.15676600

precise API calls, performing actions such as sending emails, recovering input tray content or applying labels and filters. In this paper, we carry out an evaluation of a use case of different LLM by integrating them with an email

administration system through API tools calls. Our goal is to evaluate how each model interprets the indications, choose the appropriate tools and maintain the context during the interactions. We specifically analyze the behavior of the LLM in real - time scenarios that involve several steps tasks, response latency and accuracy of the decision by invoking Gmail's functions. We also explore how well these models handle the use of the dynamic tool, including their ability to recognize when a function is needed, format the required parameters and adapt according to changes in feedback or context

When studying the performance and reliability of the LLM in this configuration of calls of functions, our research provides practical information on the viability of the LLM as agents of autonomous tasks. This exploration is crucial to understand the preparation of current models that will be integrated into complex workflows and highlights the importance of the robust mechanisms of tool calls to allow real world applications. In addition, the findings of this work can inform the development of future systems and tools designed to operate efficiently and safely within services -based architectures.

II. **Related Work**

The integration of large language models (LLM) with tools that request smart emails is based on significant advances in the processing of natural language (NLP) and automation. Several studies have explored the capabilities of the LLM in the understanding, generation and interacting with textual data, forming the basis of the email solutions promoted by the AI.

Vaswani et al. (2017) [1] introduced the architecture of the transformer, which revolutionized the NLP by allowing efficient learning based on attention. This architecture supports modern LLMs, improving its ability to process long sequences, a crucial aspect of summarizing and organizing emails.

Radford et al. (2019) [2] developed GPT-2, demonstrating the effectiveness of the previous prison not supervised in the generation of text. His work laid the foundations to use LLM in the automation of the email response, which shows how AI can generate consistent and contextually relevant responses.

Brown et al. (2020) [3] extended this research with GPT-3, which exhibited learning capabilities of few superior shots. This advance is particularly relevant for email categorization and the generation of responses, which allows AI systems to adapt to user preferences with minimal supervision. Schick and Schütze (2021) [4] explored advanced learning for text classification, which shows how LLMs can be adjusted for specific domains. Their findings are applicable to the email classification, where AI can intelligently classify messages based on priority, intention and content.

Ouyang et al. (2022) [5] instructed GPT, an adjusted version of GPT-3 that incorporates human feedback for a more conscious text generation of the context. This research supports the personalization of the email promoted by the AI, ensuring that the answers are aligned with the user's specific communication styles.

Bomma Sani et al. (2021) [6] examined the base models and their potential for real world applications. They highlighted how LLMS can interact with exterior APIs to perform tasks beyond the generation of text, a critical component of the tool that requests automation by email.

DOI: https://doi.org/10.5281/zenodo.15676600

Zhang et al. (2023) [7] investigated the integration of LLM with calls from API - based tools, demonstrating how AI can interact with external applications such as calendars, task managers and CRM systems. Its findings provide information on how LLM can schedule meetings, summarize emails and automate repetitive workflows, making email administration more efficient. Zhou et al. (2024) [8] introduced Nexus Raven, a commercially designed open permissive -designed open -sized source language model for calls for robust and precise functions. The authors emphasized the high precision of the model in the generation of structured outputs, so it is ideal for real world applications such as API invocation. Its findings are directly applicable to the email systems promoted by LLM, where the model can reliably call Gmail's APIs to compose, send or administer emails using structured tools.

Liu et al. (2024) [9] introduced Hool Ace; An automated pipe designed to generate various high quality training data to improve the calling capabilities of functions in large language models (LLM). When healing an integral group of more than 26,000 API and using a double layer verification system, Tool Oce guarantees the precision and complexity of the synthesized data. Its approach is applicable to tasks such as email automation, where AI systems can effectively interact with APIs to administer and send emails based on user instructions.

Song et al. (2025) [10] introduced the Navi so called, a reference point designed to evaluate large language models (LLM) in its ability to handle complex tasks of API functions, including the selection of extensive lists, sequence execution and called API nested. Their findings are applicable to email automation, where AIS must select and invoke the API related to appropriate email to administer tasks, such as sending, organizing and recovering electronic emails.

These studies collectively provide the basis for integrating LLM with tool calls to improve automation and email administration. By taking advantage of the NLP of the latest generation and the interactions of external tools, this research aims.

.

III. Methodology

This section describes the approach adopted to evaluate the calls of calls for functions of large language models (LLM) within the context of automation of the email workflow. The methodology includes the formulation of
use cases, system integration, models selection and evaluation configuration.

A. Use Case: Email Workflow Automation

To evaluate functions calls in a real-world configuration, the task of automating email workflows is selected. The case of use includes three categories: (1) email composition, including the writing of context -based responses and summaries; (2) Email reading, implying analysis, prioritization and extraction of key information; and (3) email, which requires the construction and sending of emails through structured API calls. These tasks emulate common interactions in productivity tools assisted by AI.

B. Each LLM is evaluated using a standardized call function consisting of the following stages:

- Prompt Input: The model receives a natural language instruction (for example, "responding to this message with a meeting playback").
- Function Schema Matching: The model identifies and populates a function call template according to predefined arguments specifications.
- API Invocation: The generated function call is executed at an end point of real email automation in a sandbox environment.
- Output Logging: The execution results and records are collected for later analysis. Langchain is used as the orchestration frame to interact with LLM with schemes of functions and routing executions to final points.

DOI: https://doi.org/10.5281/zenodo.15676600

C. Models Evaluated

The following LLMs are included in the study:

- Gemini 2.5 Pro (Preview)
- Qwen 72B Instruct
- Deep Seek Chat v3-0324
- Llama 3.3 70B
- Gpt-4o

D. Evaluation Criterion

The primary evaluation metric is the precision of invocation of functions, defined as the proportion of generated function calls that are syntactically valid and semantically aligned with the planned task. This metric captures the model of the model to interpret the user's intention and translate it into executable API calls that meet the functions scheme. Synthetic email data sets are used to simulate realistic input scenarios while avoiding the use of confidential data. All evaluations are carried out at real API points configured in a safe non - production environment.

IV. Result

The function calling accuracy of five large language models (LLMs) was evaluated on a benchmark task involving email workflow automation using real-world API calls. The results are presented in Fig. 1, which illustrates the percentage of correct function calls produced by each model.

Among the models tested, GPT-4o achieved the highest accuracy at 92.3%, demonstrating superior capability in understanding prompts and selecting the appropriate functions. Qwen 72B Instruct followed with an accuracy of 88.5%, indicating strong performance close to that of GPT-4o. DeepSeek Chat v3-0324 achieved 85.7%, slightly ahead of LLaMA 3.3 70B at 83.2%. Gemini 2.5 Pro, while still performing at a competent level, exhibited the lowest function calling accuracy at 80.4%.

These results suggest that instruction-tuned models with robust few-shot or zero-shot reasoning capabilities perform better in structured task automation scenarios. GPT- 4o's performance indicates its reliability for real-world deployment in email agent systems, while the relative rankings of the other models provide a baseline for future improvements and fine-tuning.

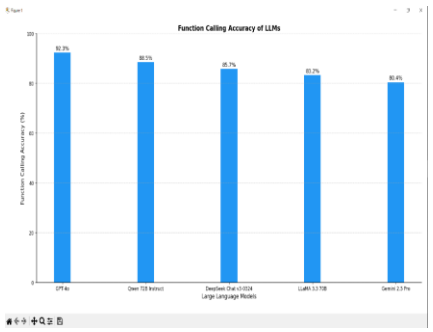

Fig. 1. Function Calling Accuracy of different LLMs.

V. Conclusion

This work introduces an intelligent email management system that combines Large Language Models (LLMs) with the Gmail API using function calling mechanisms. The system enables users to interact with their email inbox through natural language commands, automating operations such as reading, composing, replying to, and organizing emails. By exposing Gmail functionalities as tools callable by LLMs, the approach significantly reduces manual effort and enhances user productivity.

A comparative evaluation of five leading LLMs—GPT-4o, Qwen 72B Instruct, DeepSeek Chat v3-0324, LLaMA 3.3 70B, and Gemini 2.5 Pro—was conducted to assess their effectiveness in function calling. As depicted in Fig. 1, GPT-4o demonstrated the highest accuracy, followed closely by Qwen and DeepSeek Chat, with all models showing competitive performance. These findings support the viabilityof LLM-based agents in executing structured API-driven tasks within dynamic real-world environments.

The outcomes of this project underscore the transformative potential of agentic AI systems in simplifying digital workflows. As future work, the system can be extended to support multimodal interaction, cross-platform email providers, and deeper personalization. Ultimately, this research offers a scalable and generalizable foundation for AI-driven

DOI: https://doi.org/10.5281/zenodo.15676600

productivity tools in both personal and professional domains.

References

[1]A. Vaswani, N. Shazeer, N. Parmar, et al., "Attention is all you need," in Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NeurIPS), Dec. 2017, pp. 5998–6008.

[2]A. Radford, J. Wu, R. Child, et al., "Language models are unsupervised multitask learners," OpenAI Tech. Rep., Feb. 2019.

[3]T. Brown, B. Mann, N. Ryder, et al., "Language models are few-shot learners," in Proc. 34th Conf. Neural Inf. Process. Syst. (NeurIPS), Dec. 2020, pp. 1877–1901.

[4]T. Schick and H. Schütze, "Exploiting cloze questions for few-shot text classification and natural language inference," in Proc. 16th Conf. Eur. Chapter Assoc. Comput. Linguist. (EACL), Apr. 2021, pp. 255–269.

[5]L. Ouyang, J. Wu, X. Jiang, et al., "Training language models to follow instructions with human feedback," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), Dec. 2022.

[6]S. Bomma Sani, D. A. Hudson, E. Adeli, et al., "On the opportunities and risks of foundation models," Center for Research on Foundation Models (CRFM), Stanford University, Aug. 2021.

[7]Y. Zhang, L. Zheng, J. Yan, et al., "Towards tool-augmented LLMs: Integrating APIs for enhanced task automation," in Proc. 61st Annu. Meet. Assoc. Comput. Linguist. (ACL), Jul. 2023.

[8]Y. Zhou, R. Chen, D. Zhang, et al., "Nexus Raven: A commercially permissive language model for robust function calling," in Proc. IEEE Conf. Big Data (Big Data), Dec. 2024.

[9]Y. Liu, M. Chen, J. Huang, et al., "Tool ACE:

[10]Winning the points of LLM function calling," in Proc. 2024 Int. Conf. Artificial Intelligence (ICAI), Sep. 2024.

[11]Z. Song, W. Li, M. Yu, et al., "Call Navi: Evaluating large language models for complex API function calling," in Proc. 2025 Conf. Empirical Methods in Natural Language Processing (EMNLP), Jan. 2025.

[12]J. Wulf and J. Meierhofer, "Exploring the potential of large language models for automation in technical customer service," in Proc. 2023 Int. Conf. Service Science (ICSS), May 2023.

[13]Z. Wu, H. Gao, J. He, and P. Wang, "The dark side of function calling: Pathways to jailbreaking large language models," in Proc.

[14]1st Int. Conf. Computational Linguistics (COLING), Jan. 2025, pp. 584–592.

[15]Y. Zhang, L. Zheng, J. Yan, et al., "Facilitating multi-turn function calling for LLMs via compositional task planning," in Proc. 2024 Conf. Empirical Methods in

Natural Language Processing (EMNLP), Dec. 2024.

[16] In Gim, Seung-seob Lee, and Lin Zhong, "Asynchronous LLM Function Calling," in Proc. 2024 Conf. Neural Inf. Process. Syst. (NeurIPS), Dec. 2024.

[17] Varatheepan Paramanayakam, Andreas Karatzas, Iraklis Anagnostopoulos, and Dimitrios Stamoulis, "Less is More: Optimizing Function Calling for LLM Execution on Edge Devices," in Proc. 2024 Int. Conf. Edge Computing (EDGE), Nov. 2024.

[18] Graziano A. Mandazi, Federico A. Galatioto, Mario G. C. A. Cimino, et al., "Improving Small-Scale Large Language Models Function Calling for Reasoning Tasks," in Proc. 2024 Int. Joint Conf. Artificial Intelligence (IJCAI), Oct. 2024.

DOI: https://doi.org/10.5281/zenodo.15676600

[19]M. Kulkarni, "Agent-S: LLM Agentic Workflow to Automate Standard Operating Procedures," in Proc. 2025 Int. Conf. on Artificial Intelligence (IJCAI), Feb. 2025.

[20]A. Sharif, "LLM-Based Email Responder: A Fast API Implementation to Generate Response Emails Considering Context and Tone," in Proc. 2024 Int. Conf. on Natural Language Processing and Applications (NLP-A), Jul. 2024.

[21]Khare, S. Singh, R. Mishra, et al., "E-Mail Assistant: Automation of E-Mail Handling and Management using Robotic Process Automation," in Proc. 2022 Int. Conf. on Intelligent Computing and Control Systems (ICICCS), May 2022.

[22]J. Thier Gart, S. Huber, and T. Uebelacker, "Understanding Emails and Drafting Responses -- An Approach Using GPT-3," in arXiv preprint arXiv:2102.03062, Feb. 2021.

DOI: https://doi.org/10.5281/zenodo.15676600