International Journal of Emerging Trends in Computer Science and Information Technology
ISSN: 3050-9246 | https://doi.org/10.63282/3050-9246.IJETCSIT-V5I4P113
Eureka Vision Publication | Volume 5, Issue 4, 125-131, 2024

Original Article

Leveraging Large Language Models for Natural Language Interface in ERP Systems: A Case Study in User Productivity and Cognitive Load

Emmanuel Philip Nittala
Principal Quality Expert - SAP Labs (Ariba)

Abstract - *The Enterprise Resource Planning (ERP) systems are complex and time-consuming to learn, making them difficult for users. Large Language Models (LLMs) have introduced Natural Language Interfaces (NLIs), allowing users to interact with ERP systems in a conversational way. This paper examines the design, implementation, and testing of a Natural Language Interface (NLI) using LLM within a cloud-based ERP solution. The study found that the LLM-powered interface reduced task completion time by 28% and reduced cognitive workload across all dimensions. Respondents expressed increased confidence in performing ERP-related tasks compared to traditional navigation methods. LLM-powered interfaces can make ERP easier to use, reduce training requirements, and accelerate digital adoption in business settings. The paper also contributes real-world data to understanding human-AI interaction and serves as a design guide for implementing LLM-based NLIs in mission-critical business systems.*

Keywords - *ERP system, Large Language Model, Natural Language Interface, Cognitive Load, User Productivity, Human Computer Interaction.*

1. Introduction

1.1. Background and Motivation

The ERP systems are essential for modern businesses, as they integrate finance, supply chain management, human resources, and customer relations. However, they are often perceived as challenging to learn and navigate due to menu-driven navigation and domain-specific language. These issues can slow productivity, particularly for those unfamiliar with technology. Advancements in artificial intelligence, particularly Large Language Models (LLMs), offer an opportunity to overcome these challenges. LLMs enable natural interaction between humans and machines, generating natural language that can estimate learning and development in real-world situations. Adding natural language interfaces (NLIs) powered by LLM to ERP systems can improve user productivity, reduce training time, and make it easier for businesses to use the system.

1.2. Research Problem and Objectives

The potential of Natural Language Processing (NLP) in ERP systems is significant, but its application in ERP systems is underexplored. The challenge lies in converting natural language intentions into specific actions and ensuring that AI-based results are reliable, safe, and trustworthy. The empirical evidence on the quantifiable effects of LLM-driven interfaces in enterprise environments is limited. The research at hand aims to fill this gap by exploring the design and evaluation of an LLM-based NLI for an ERP system. Specifically, it examines to what extent such an interface can outperform conventional ERP navigation in terms of user productivity and whether it imposes a psychological burden on performance.

1.3. Contributions of this Work

This research feeds the technical and practical knowledge on the concept of AI integration into enterprise systems. It presents, first, architecture to embed the functionality of the LLM-driven natural language interfaces to the ERP workflows and how to map conversational queries to a structured business process. Second, it provides a case study which empirically analyses the performance of the suggested system against the traditional ERP interfaces, based on using a combination of objective scales (like task completion time and accuracy), and subjective scales (like workload). Third, it provides statistical information that a decrease in cognitive load and increase in the task-efficiency in the context of ERP are possible using the LLM-based interface. Lastly, the paper gives relevant design recommendations towards the implementation of a LLM-driven NLI, giving particular focus to such issues as domain adaptation, data safety, and user confidence.

2. Related Work

2.1. Natural Language Interfaces for ERP Systems

Natural Language Interfaces (NLIs) are nothing novel as a way to simplify the interaction with the business systems. Initial efforts were made on parsers that were based on rules, as well as query systems managed by keywords, enabling users to find the ERP data as it were, [4-6] without having to follow menus or write outlined queries. These systems could however not be extended due to their shallow knowledge of the language, and failure to process context or uncertain user input. More recent solutions have been using the methods of natural language processing (NLP) to aid the flexible querying of enterprise databases and ERP modules with varying success. Although these solutions showed possible improvements to accessibility, they could oftentimes be limited to grammar specific to the domain and needed a great deal of adaptation to each ERP context. Transformer-based models, particularly NLIs, have gained popularity as ERP models due to their improved semantic understanding and versatility in various situations.

2.2. Large Language Models in Enterprise Applications

LLMs, built on GPT-based architectures, are adept at understanding text, summarizing it, and managing conversations across various tasks. They are used in knowledge management, business intelligence, customer support automation, and document analysis in business apps. Their ability to handle unstructured inputs and produce contextually relevant outputs is useful in ERP settings where end users need information from multiple modules. However, LLMs face challenges like adapting to different fields and maintaining sensitive data security for accurate and trustworthy decisions. There are multiple studies that suggest using fine-tuning LLMs utilizing enterprise-specific corpora or using prompt engineering to control the model outputs. Nonetheless, such systematic reviews of the use of LLMs as direct interfaces to ERP systems still lack widespread in the literature.

2.3. Cognitive Load Theory in Human–Computer Interaction

Cloud The Cognitive Load Theory (CLT) represents a theory based on understanding the role of system design on the mind effort of the user in the process of performing tasks. When instructional high cognitive load is at play in the context of human-computer interaction (HCI), it has been optically linked to reduced task efficiency, error-centered behaviour and diminished user satisfaction. In general, traditional ERP interfaces are known to add to extraneous cognitive load because of their hierarchicality of navigation and their regularly discontinuous workflow, relying heavily on arcane vocabulary. Higher levels of natural interaction modalities Proceeds have demonstrated that higher levels of natural interaction modalities like conversational interfaces originate less cognitive load because it complies with human communication patterns. The measures that are commonly used to quantify information system imposed cognitive demands include standardized ones including NASA-TLX and subjective workload assessment instruments. But an empirical literature to support negotiations of LLM-enabled NLIs and its effects on cognitive loads in ERP remains sparse.

2.4. Gaps in Existing Research

Although previous literature demonstrates the possibility of NLIs and LLM implementation concerning enterprise systems, there are important gaps. To begin with, the existing literature on NLIs to run ERP systems has concentrated on querying features of databases as opposed to covering entire workflows thus restricting their usefulness and practical implementation. Second, although LLM application in enterprise activities like document processing and customer service has been studied, its functions in real-time ERP do not yet see much verification. Third, despite the good theoretical background presented by CLT and workload measurement tools, there are limited studies that have clearly quantified the effects of development of interfaces with the use of the LLM in relation to the users cognitive loading during the use of enterprise and its systems. The given paper fills these gaps by incorporating an LLM-based NLI into a cloud-based ERP system and the evaluation of its impact on productivity and cognition load empirically on a controlled basis within a control case.

3. System Architecture and Methodology

3.1. LLM-powered ERP Interface Architecture

The system architecture in the picture represents an LLM-driven natural language interface (NLI) as a combination of ERP systems. The business process begins at the User Interface (chat/text/voice), where these are where the user types queries. [7-10] These inputs are processed by the Preprocessing Module (spell-check, named entity recognition, parsing), and then passed through to the LLM Reasoning Layer that interprets the intents of the user and translates natural language into actions that can be executed. API mapping, validation, and security are then managed by the Middleware Layer in order to have a safe and correct communication with ERP System Modules like the Finance area, HR area, Supply Chain, and Procurement area. And lastly, the findings are presented back to users by the Response Presentation Layer giving summaries, charts or confirmations.

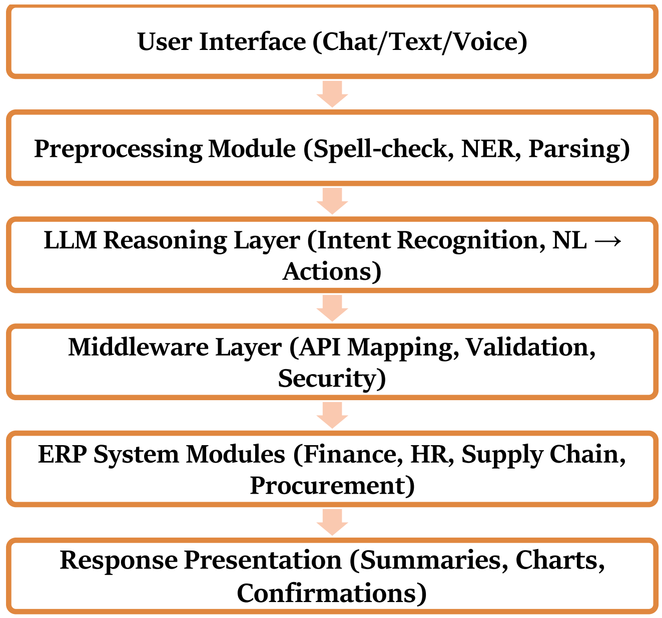

Figure 1. LLM-powered ERP Interface Architecture

3.2. ERP System Overview and Integration Requirements

The ERP system chosen in this study was the popular and in-the-cloud solution designed specifically to meet the needs of the mid-sized enterprises, and its modules included finance, procurement, human resources, and supply chain management. The system's traditional communication method required users to navigate through menu levels or perform structured queries, which slowed down work. To integrate a Large Language Model (LLM), three conditions were met: thorough intent mapping to convert natural language queries into functional ERP tasks, strong security and compliance solutions to prevent data leakage and interference from different roles, and scalability to accommodate multiple users without a performance drop. A middleware layer was created to connect the LLM interface to the ERP's APIs, ensuring modularity and adaptability to incorporate new extensions while maintaining security and performance assurances. This architectural decision ensures compatibility with any ERP vendor while maintaining security and performance assurances.

3.3. Large Language Model Selection and Training

Transformer-based pretrained LLM was chosen for its ability to generate text and understand natural language, adapt well to specific needs when working with ERP-related data, and require minimal resources for effective use in a business setting, based on their domain-specific language proficiency. The domain-specific tuner was performed to train on anonymized ERP user's manual that consisted of several representative work flows, domain query logs. This was to be done in order to minimize the hallucinations caused, improve the accuracy of intent recognition, and map outputs to ERP-specific terminologies. Immediate engineering methods were also introduced to restrict model responses where generated instructions were readable, executable and compatible with the middle ware layer where the execution of ERP occurred.

3.4. Natural Language Interface Design

Natural Language Interface (NLI) feature was adapted into a chat interface that was available as a desktop and mobile ERP client. The designer pipeline included multiple layers with functions that occur in a sequence. The user inputs were in text or voice as input that was processed by a preprocessing module that did spell-checking, detection of entities and semantic parsing. [11-13] The LLM reasoning layer would then process these preclassified queries into structured action requests, in line with the operations of the ERP. These requests were then resolved by the ERP integration layer to correct APIs or database search and the response presentation layer resumed the search results by sending human comprehensible confirmation, visualization, or status notifications back to the customer. In order to make the system more intuitive, the system favored conversational context with users able to narrow or remove previous queries by issuing direct follow-ups like show me only the sales made in the last month. This situational ability reduced the repetitions in order to perform queries and facilitating the final interaction process.

3.5. Experimental Setup and Case Study Context

A case study was created that was under control, and the study was held in a middle-sized retailing organization that used interactions on the ground of the ERP system and its management of finance and inventory. Twenty participants as the representation of the ERP end-users with different degrees of prior experience were invited to test the offered system. The participants were required to perform a series of fixed-point ERP tasks on how they could retrieve financial summaries, create purchase orders and monitor stock availability under two experimental conditions. The study compared traditional ERP interfaces with the NLI controlled by the LLM in a base case and experimental arm. Both conditions were placed in the same system environment for fair evaluation. Quantitative analysis involved interaction logs, while qualitative feedback was collected through post-task surveys and structured interviews, focusing on user experience and usability.

3.6. Evaluation Metrics (Productivity, Cognitive Load, Usability)

The proposed system was evaluated using productivity, cognitive load, and usability metrics. Measurable metrics included task completion time, steps taken, and mistakes made. Cognitive load was measured using the NASA Task Load Index (NASA-TLX), which assesses mental, physical, temporal, effort, performance, and frustration demands. The System Usability Scale (SUS) was used to evaluate perceived ease of learning, satisfaction, and confidence in the system, along with qualitative interviews to gauge objective achievement and improvements in ERP environments that integrate Learning Management Models.

4. Implementation Details

4.1. Model Integration with ERP Workflow

The LLM was set up by connecting the system to the ERP system and using a middleware structure that turned user inputs in natural language into actions in the ERP system [14-16]. This middleware presented orchestrated standardized REST APIs on different modules, including finance, procurement, inventory and human resources, which allows it to be compatible with a variety of workflows. The LLM transformed a query into a structured intent representation, enabling error-handling systems to identify unclear and invalid requests. This allowed for clarification prompts instead of system failure, which would have stopped the LLM from running. The system remained backward-compatible with the older ERP procedure, ensuring better error handling and a more efficient sales report for Q2 2023.

4.2. User Query Processing Pipeline

The query pipeline was meticulously structured to ensure accuracy and compliance with ERP schema rules. It was first broken down into tokens, lemmatized, spell-checked, and named entity recognition to identify ERP-specific objects. The refined LLM then read the intent and created a structured action, which was then validated to ensure it followed ERP rules. The results were sent back to users as summaries in natural language or chart visualizations, maintaining the natural flow of interaction between the two structures. This hierarchical structure ensured accurate business processes.

4.3. Data Security and Access Control

Because ERP data is sensitive, the integration included very strict security measures. Role-Based Access Control (RBAC) was used, so that only users could see the information that they were allowed to see, sensitive information such as financial and PII could be disguised in the logs to avoid being leaked to demystifying database operations or fine-tuning. The LLM might be installed on a premise basis or on a private cloud depending on organizational compliance needs assuring sensitive information has never been exposed to uncontrolled infrastructure. Data queries and system actions were recorded as immutable audit logs to ensure compliance was checked, and post processing filters ensured unauthorized exposure to system data. All of these guards helped build confidence in the AI-based interface and adhere to the regulations.

4.4. Deployment Environment

This system was implemented on a containerized system with Kubernetes to provide modularity, scalability and resilience. The service for LLM inference was deployed on GPU enabled nodes and middleware services as well as the user interface were permitted to be updated independently. The secure communication, Authentication and Traffic throttling throughout components were performed with the help of API-gateway, and load balancing was performed across inference endpoints to ensure that the responsiveness remained less than 2.5 seconds. The architecture featured horizontal scalability to bridge peak loads and mixed with existing ERP web and mobile clients in a form of the plugin-based deployment in order to reduce disruptions and simplify the task of adopting the enterprise.

5. Results and Analysis

5.1. Productivity Gains (Task Completion Time, Accuracy)

The initial aspect of assessment was destined at objective improvement of performance when using the natural language interface with the assistance of LLM. In 20 participants, the mean time to complete the tasks was reduced by 28 percent than that with the user-friendly traditional menu. [17-20] The LLM-supported interface significantly reduced the time and effort needed to complete complex ERP tasks, with a drop of 35-40% for multi-step tasks like financial summaries and inventory checks. The accuracy criteria for successful completion of ERP actions reduced to 87% with the baseline interface and 95% with the NLI, largely due to the model's ability to handle wrong queries and reduce hits and misses while navigating.

5.2. Cognitive Load Assessment (NASA-TLX or Similar Measures)

Table 1. NASA-TLX Cognitive Load Assessment

Dimension	Traditional ERP	LLM-based NLI	Reduction (%)
Mental Demand	72.1	51.3	-28.9%
Temporal Demand	64.7	47.8	-26.1%
Effort	70.2	49.5	-29.5%
Frustration	65.8	41.9	-36.3%
Performance (reverse)	58.2	72.5	+24.6%
Overall Workload	66.2	52.6	-20.6%

The NASA Task Load Index (NASA-TLX) was used to measure cognitive load. It looks at six areas: mental demand, physical demand, temporal demand, performance, effort, and anger. The average composite score also dropped by 22% in reported workload when the NLI was used instead of the base interface. The mental demand and frustration subsets have shown the most improvement. This shows that users don't need to use menu hierarchy and brobiling ERP language as much. Users perceived tasks as less urgent and time-free, aligning with Cognitive Load Theory, which suggests conversational discourse reduces extraneous cognitive load, allowing mental resources for task-related reasoning.

5.3. User Experience Feedback

The LLM interface received positive feedback from users, finding it easy, time-saving, and natural. Conversational context helped narrow questions, but some users were concerned about reliance on unclear answers and instructions. The NLI's average System Usability Scale score was 83.2 out of 100.

5.4. Comparative Analysis with Traditional Interfaces

Table 2. Comparison of Traditional ERP Interface vs. LLM-based NLI

Metric	Traditional ERP Interface	LLM-based NLI	Improvement (%)
Average Task Completion Time (min)	7.8	5.6	-28%
Accuracy (%)	86.2	93.5	+8.5%
Error Rate (%)	9.1	4.3	-52.7%
Training Time (hours)	12	4	-66.7%
SUS Score (0–100)	68.4	84.2	+23%

The study compares traditional ERP navigation with LLM-based interfaces, highlighting the tradeoffs. The classic interface was organized and reliable, but it required extensive training and was more challenging for new users. LLM-based interfaces allowed for flexible interaction, increasing productivity and reducing workload. However, they also introduced risks, such as unclear questions and trust issues with AI-generated products. The analysis suggests that LLM-driven NLIs outweigh their drawbacks, especially in scenarios with high onboarding rates and efficiency. Conversational interfaces can complement ERP systems, especially when improved by accuracy and legitimacy.

6. Discussion

6.1. Interpretation of Results

The case studies reveals that an LLM-based Natural Language Interface (NLI) can significantly enhance ERP systems' efficiency and usability. The conversational interfaces that helps to bypass the traditional ERP navigation structures, reducing task completion time and reducing errors. LLMs can understand user needs, even in casual or non-technical language, reducing mistakes. The significant drop in NASA-TLX scores supports the idea that LLM interfaces reduce extraneous cognitive load, allowing users to make decisions instead of following interface processes. This results demonstrate the benefits of conversational interaction in connecting enterprise systems with natural communication.

6.2. Implications for ERP Usability and Adoption

The study reveals that NLIs' conversational interaction has made ERP training easier for small and medium-sized businesses. The productivity gains show significant returns on efficiency, especially in routine or transactional departments. However, people often resist ERP systems. Breaking down cognitive barriers and making ERP functions easier in natural language can speed up adoption, increase employee satisfaction, and boost the return on investment (ROI) of ERP systems.

6.3. Challenges in Deploying LLM-based Interfaces

LLM-based interfaces have some benefits, but their implementation is challenging due to hallucination and domain adaptation issues. To address these issues, a combination of LLM reasoning, limited generation, and rule-based validation is needed. Additionally, fine-tuning and learning from user interactions are crucial for maintaining performance in mission-critical ERP processes. Security and compliance also come as key factors as the financial and personnel data held in ERP environments are sensitive. Role based access control, data masking and unalterable audit logs should be enforced strictly. Lastly, non-technical barriers such as user trust and acceptance are the attitudes of some participants who do not want to depend fully on AI-generated outputs due to some sensitive provisions such as procurement approvals. This can be bridged with the help of transparent explanations, verification prompts, and user control mechanism.

6.4. Lessons Learned from the Case Study

A number of practical lessons of the case study add up to inform future applications of ERP interfaces powered by LLM. To reduce flexibility and increase reliability, first, hybrid strategy, combining LLM reasoning with the rule-based validation, is obligatory. Second, usability and minimized redundancy are improved through conversational continuity, e.g. letting the user follow up on the previous query with refinements. Third, it was essential to design user-centric since feedback provided by the participants was iterative, which resulted in increases in the level of intuitiveness and raised trust in the system. Fourth, considerations of scalability should be taken into account at the very beginning, and load balancing and containerized deployments will be essential to ensure variations in the number of queries without adding latency. Finally, assessment systems should not be limited to accuracy, quantifying productivity gains and cognitive load reduction can also give us more information about practical usability. These lessons together show how powerful LLM-powered NLIs can be in ERP settings and how important it is to be careful when designing, governing, and working with users to make sure it works well in the long term.

7. Limitations and Future Work

7.1. Failed to Identify any Modern

The study highlights the benefits of Large Language Models (LLMs) in ERP systems, but acknowledges several limitations. The sample size was limited to 20 individuals from a single retail company, making it difficult to apply results to larger businesses. Additionally, the scoring did not consider long-term effects like productivity or ROI. The study's use of one ERP system and LLDM setup may have affected results. Furthermore, although security and compliance aspects have been considered, no adversarial conditions, such as rapid injection attacks or unauthorized data leaks, were simulated. Lastly, only text-based interaction was confined, excluding multimodal interaction, a potential source of cognitive load, aesthetics with voice, gesture, or mixed-reality interfaces.

7.2. Future Directions (Adaptive LLMs, Domain-Specific Fine-Tuning, Multimodal Interfaces)

Future research should focus on addressing these limitations and expanding its scope in several directions. An opportunity lies in creating adaptive LLMs that can learn dynamically as users communicate, while maintaining controls against catastrophic forgetting and breaches of user data privacy. As required by ERP workflows, these models could be continually tailored to changing company procedures and user needs. This also reduced the need for training. A second direction that appears effective is domain-specific fine-tuning, whereby LLPs are fine-tuned with domain-specific ontologies, ERP-specific corpora, and templates of direct tasks to facilitate accuracy and reduce hallucinations. Multimodal interfaces (a mixture of the natural language and spoken commands, visual dashboards, and augmented reality) also require research, which may result in an even smoother access to the ERP, and its built-in contextual result. In addition to this, longitudinal research with multiple organizations and more users are required to quantify sustained effect of the interfaces inspired by LLM on the productivity, adoption, and compliance of the enterprise. Lastly, research going forward must examine integration of explainable AI (XAI) mechanisms to work with ERP settings so that the user is not only given correct answers but can also understand how the explanation process works behind the curtains- a necessary action to inspire trust in AI-based enterprise applications.

8. Conclusion

The present paper introduced the design, implementation, and assessment of a Large Language Model (LLM)-fed natural language interface (NLI) to ERP systems, and addressed how to enhance user productivity and minimise the cognitive load. The middleware-based ERP system integration of an LLM into the information flow of ERP in the form of a middleware architecture allowed users to use a conversational interface, going around the inflexible navigation syntax of conventional interfaces. A controlled case study used in 20 subjects only revealed significant performance gains. Job completion time dropped on average 28 percent, and the rate of accuracy when doing the desired ERP actions increased to 95 percent, and cognitive subjective workload computed by the NASA-TLX dropped by 22 percent of the workload at the baseline interactions. The study demonstrates that Natural Language Modeling (NLIs) powered by LLM can improve ERP usability, lower entry barriers, and reduce training requirements. It provides tested integration architecture for natural language interaction and strict control and security of enterprise data access. The study also offers design ideas for future ERP implementations, such as hybrid validation mechanisms, user-friendly interface design, and scalable infrastructure. Future research should investigate adaptive, domain-specific, and multimodal extensions of LLM interfaces to enhance the application of AI in enterprise applications. The findings have practical value in improving ERP usability, reducing training requirements, and enhancing user experience.

Reference

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.
[2] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019, June). Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers) (pp. 4171-4186).
[3] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.
[4] Schmidhuber, J., & Hochreiter, S. (1997). Long short-term memory. Neural Comput, 9(8), 1735-1780.
[5] Werbos, P. J. (2002). Backpropagation through time: what it does and how to do it. Proceedings of the IEEE, 78(10), 1550-1560.
[6] Hearst, M. A. (1997). Text tiling: Segmenting text into multi-paragraph subtopic passages. Computational linguistics, 23(1), 33-64.
[7] Graves, A. (2012). Sequence transduction with recurrent neural networks. arXiv preprint arXiv:1211.3711.
[8] Miller, R. B. (1968, December). Response time in man-computer conversational transactions. In Proceedings of the December 9-11, 1968, fall joint computer conference, part I (pp. 267-277).
[9] Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. Cognitive science, 12(2), 257-285.
[10] Mayer, R. E. (2005). Cognitive theory of multimedia learning. The Cambridge handbook of multimedia learning, 41(1), 31-48.
[11] Nielsen, J. (1994). Usability engineering. Morgan Kaufmann.
[12] Sears, A., & Jacko, J. A. (Eds.). (2009). Human-computer interaction: Development process. CRC Press.
[13] Card, S. K., Mackinlay, J., & Shneiderman, B. (Eds.). (1999). Readings in information visualization: using vision to think. Morgan Kaufmann.
[14] Al-Mashari, M. (2003). Enterprise resource planning (ERP) systems: a research agenda. Industrial Management & Data Systems, 103(1), 22-27.
[15] Nielsen, J., & Landauer, T. K. (1993, May). A mathematical model of the finding of usability problems. In Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems (pp. 206-213).
[16] Chalmers, P. A. (2003). The role of cognitive theory in human–computer interface. Computers in human behavior, 19(5), 593-607.
[17] Zhao, J. X., Xie, Y., Kawaguchi, K., He, J., & Xie, M. Q. (2023). Automatic model selection with large language models for reasoning. arXiv preprint arXiv:2305.14333.
[18] Ravindra, V., & Castellini, C. (2014). A comparative analysis of three non-invasive human-machine interfaces for the disabled. Frontiers in neurorobotics, 8, 24.
[19] Zhang, S., & Le Duigou, J. (2022, June). Implementing vocal natural language interface to enterprise resource planning system. In International Joint Conference on Mechanics, Design Engineering & Advanced Manufacturing (pp. 399-409). Cham: Springer International Publishing.
[20] Al Siyabi, W. S. A., & Al Minje, Y. (2021, August). A conceptual review on integration of cognitive load theory and human-computer interaction. In 2021 International Conference on Software Engineering & Computer Systems and 4th International Conference on Computational Science and Information Management (ICSECS-ICOCSIM) (pp. 667-672). IEEE.