Modular Architecture and Intelligent Routing for Chatbots

Claudiano Leonardo da Silva[1], Bruna Alice Oliveira de Brito[2], Sérgio Natan Silva[3], João Vítor Venceslau Coelho[1], Jean Mário Moreira de Lima[2(✉)], André Morais Gurgel[2], and Itamir de Morais Barroca Filho[2(✉)]

[1] Departamento de Engenharia de Computação e Automação - UFRN, Natal, Brazil
[2] Instituto Metrópole Digital - UFRN, Natal, Brazil
jean.lima@imd.ufrn.br
[3] Unidade Acadêmica de Engenharia Elétrica, UFCG, Campina Grande, Brazil

Abstract. With the growth and popularity of large-scale language models (LLMs), chatbots have become essential to transforming interactions between business users through specialization, personalization, efficiency, and scalability. Nevertheless, issues such as hallucinations, specialization limits, and maintaining accuracy while multitasking remain critical challenges. In response to these concerns, this paper suggests a modular architecture for smart chatbots with LLMs as dynamic routers to route user queries to expert agents. The idea is to reduce hallucinations while increasing the relevance and accuracy of responses and show the strength of more specialist prompts. The suggested architecture classifies agents by domain (health and finance) and then by subdomains like symptoms or money planning. More advanced prompting techniques, including chain-of-thought and few-shot learning, are employed to improve contextual understanding and response accuracy. Furthermore, the findings indicate that using different language models for different situations can maximize query routing and maximize the level of detail in responses, which will be better matched to user needs. Experimental results confirm that the modularity-based solution is viable under multi-domain conditions and reduces hallucinations significantly while enhancing outputs accuracy and contextualization.

Keywords: chatbots · generative IA · language models · dynamic routing · modularity

1 Introduction

The association between big data and progress in machine learning has changed how businesses operate across many industries [14]. With AI being adopted more widely, digital transformation is reshaping how companies deal with customers,

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
O. Gervasi et al. (Eds.): ICCSA 2025, LNCS 15649, pp. 268–284, 2025.
https://doi.org/10.1007/978-3-031-96997-3_17

making their processes faster, scalable, and more tailored [1]. AI-powered chatbots, in particular, have become a key tool to improve customer engagement and simplify how businesses and clients interact [9].

Chatbots, which engage users through text or voice, have become important tools for delivering personalized customer experiences, especially in service-oriented environments [5]. Using personalized journeys where these generative AI solutions give companies a competitive advantage in areas such as e-commerce, customer service, and other support functions [3,6,19]. Studies show that personalization strongly impacts buying behavior, with about 80% of consumers more likely to engage or purchase when it is present [16].

The integration of generative AI models into conversational agents has recently marked a pivotal increase in their capabilities. Generative AI enhances the adaptability, responsiveness, and overall reliability of chatbots, enabling smoother and contextually aware interactions [24]. Despite all such developments, there still is a large lacuna in literature regarding a robust architectural model that not only addresses precision and confidence but also addresses important customer experience factors such as scalability, performance, maintainability, and the dynamic changing of context while communicating [11].

To address these limitations, we introduce a modular architecture for intelligent chatbots that blends dyna0mic routing using specialized agents, minimizing the dependence on large training data and preventing hallucinations in multi-tasking systems. In contrast to current solutions, our method effectively unifies multiple domains, striking a balance between scalability and maintainability.

Aside from that, the use of distinct language models for distinct circumstances increases response depth and query segmentation. The results of this study indicate that the modular approach favors agent specialization and increases contextualization to facilitate greater accuracy and appropriateness in user interactions.

The remaining paper structure is: Section 2 discusses related work and recent advancements; Section 3 approaches the proposed architecture; and Sects. 4 and 5 present our case study and experimental protocols employed to validate the solution. Finally, Sect. 6 depicts the results, with the conclusion exploring the implications and future advancements.

2 Related Work

The use of large-scale language models (LLMs) for various applications has driven significant advances in systems that seek to combine high performance and cost efficiency. Within this scenario, it can be observed that the use of retrieval-augmented generation (RAG) techniques has driven significant advances in the construction of intelligent systems, such as chatbots and personalized assistants, combining the retrieval of external information with the generation of semantically aligned content. This is evident if one considers the work developed in [17] to demonstrate that the employment of a variety of LLMs is capable of generating more uniform responses within formal business settings, whereas models

such as those developed in [2,18] use semantic alignment and question rewriting to improve the uniformity of the response. These attempts signify the importance of having specialized modules for specific tasks, such as the automation of workflows in bioinformatics or command optimization in vehicle domains.

However, with all these advances, RAG-based systems typically face challenges in terms of efficiency, scalability, and computational cost. One can see that the diversity of tasks and use cases present challenges to the adaptation of a single model, especially when trying to realize accuracy at the cost of execution costs. Under such a scenario, intelligent routing techniques, such as those based on LLM, offer a potential solution regarding the dynamic selection of committed models or agents to serve diverse queries. A variety of methodologies have, therefore, been thoroughly investigated to maximize routing performance. RouteLLM, for example, uses human preference information to train routers that decide on the best model based on query complexity and context, optimizing cost and accuracy [15].

Similarly, PolyRouter integrates multiple specialized LLMs within one system, employing methods such as MLP and BERT to learn query embeddings and predict the most appropriate model for each task. Although this approach greatly improves efficiency, it is dependent on the quality of the generated embeddings and prior training [20]. HybridRouter, on the other hand, focuses on efficient routing across the combination of models with diverse capabilities, regulating their performance based on historical data and trends discovered during training [7]. TensorRouter follows a similar direction, utilizing predictive models such as MLP and BERT to find the correlation between queries and expert LLMs. This approach requires extensive prior training to create embeddings and tune models for routing [21].

Other approaches, such as GraphRouter, utilize graph-based representations to model interactions between queries, tasks, and models, allowing for adaptive routing to new LLMs or tasks. This method also requires robust contextual data to build and refine a heterogeneous graph, as well as ongoing tuning to adapt to new models [10]. While these approaches show promising results, they also show a dependence on high-quality data for training and fine-tuning. This requirement can introduce significant overhead in the initial configuration and ongoing maintenance of the system over time, especially in dynamic configurations or with changing requirements regularly.

These works highlight the need for intelligent routing and modularity to improve accuracy, scalability, and efficiency in LLM-based systems. However, limitations such as the need for vast amounts of training data, computational costliness, and inability to adapt dynamically reinforce the argument for approaches that mitigate these impacts.

Table 1 presents the comparison of different methods of chatbots with LLMs and their strengths, weaknesses, and differences with the solution proposed in this research.

Based on this analysis, this paper proposes a modular routing architecture for intelligent routing that combines LLMs and specialized agents in a manner

Table 1. Comparison of different approaches for LLM-based chatbots.

Work	Approach	Benefits	Limitations	Difference from Our Work
RouteLLM [15]	Routing based on LLMs	Cost and accuracy optimization from human preference data	Requires high-quality human preference datasets	Our approach prioritizes flexibility and modularity
GraphRouter [10]	Graph-based routing	Scalable for integrating new LLMs dynamically	Requires robust data structuring and frequent updates	Our architecture provides greater flexibility across different domains
PolyRouter [20]	Multi-LLM querying system	Uses MLP/BERT to learn query representations and predict the best LLM	Strong dependence on embedding quality and prior training	Our model emphasizes modular specialization instead of embedding-based selection
HybridRouter [7]	Hybrid LLM routing	Combines different LLMs dynamically for cost-efficient routing	Requires extensive historical performance data for optimization	Our approach provides modular scalability without relying on prior query performance
TensorRouter [21]	Predictive model-based routing	Uses machine learning models (MLP/BERT) to match queries with the best LLM	Demands extensive pre-training to generate effective embeddings	Our architecture minimizes dependency on large-scale pre-training

that seeks to optimize chatbot response relevance and accuracy. Unlike single-model multitasking with an LLM, our method employs an LLM-based router that directs questions to the most appropriate agent, reducing hallucinations and context-personalization integration more efficiently.

The modularity of the architecture also allows for a sequence of specialized agents to be executed separately, lowering the cognitive overhead of a single model along with improving scalability. Additionally, the framework supports the addition of new domains without full retraining of the main model, making it an extremely efficient and adaptable solution to high-demand business applications.

3 Proposal

For language models to work well in specific tasks, they need access to contextual data that fits the problem. While methods like Retrieval-Augmented Generation (RAG) bring in outside knowledge to improve answers, they often struggle with multi-domain use and can cause hallucinations or lose accuracy with varied queries. In more complex scenarios or other scenarios that require more in-depth topical analysis, like chatbots that deal with both health and

finance, weak contextualization can hurt the quality of interactions and make responses less reliable.

To address this challenge, this work proposes a modular architecture, shown in Fig. 1, that uses intelligent routing and specialized agents to increase the accuracy and contextual relevance of responses from large language models. The solution reduces the reliance on large training sets and improves query handling by dividing the process into three layers: a routing module, domain agents, and expert agents. The routing module handles initial filtering, interpreting query intent, and passing it to the right domain agent. That agent then narrows the request further and sends it to a specialized agent focused on a specific topic, tuned to generate accurate answers within that area.

This module-based solution has some advantages over the usual architectures from a single multitasking LLM. Firstly, query segmentation reduces cognitive overload on the model since each specialized agent handles only a subset of questions from its specific domain. This improves response accuracy and reduces hallucinations, which is a common issue for generalist models. In addition, the modularity of the system makes it scalable to include additional domains without extensive reconfiguration or retraining of the center model. This is an important benefit in business scenarios and dynamic, rapidly changing fields where constant updating of data is a requirement.

Another unique feature of the proposal is the use of cutting-edge prompt engineering techniques to improve agent-to-agent communication and query context comprehension. Methods such as chain-of-thought prompting, which encourages the model to structure its thinking more logically, and few-shot learning, which uses examples in the prompt to guide the response, are employed to improve the coherence and relevance of dialogue. The combination of these techniques with modular design helps the system be more flexible regarding different question forms and levels of complexity, improving greater flexibility for response generation.

For the purpose of ascertaining the effectiveness of the proposed methodology, this paper utilizes a realistic dataset that includes queries in finance and healthcare scenarios. The paper attempts to compare the use of traditional methods based on a single multitask model versus the proposed system.

4 Case Study

This case study demonstrates how the proposed architecture, utilizing intelligent routing and domain-specific agents, can improve response relevance in chatbots that manage multiple domains. To simulate this scenario, two datasets were utilized: one related to financial questions [13] and another for the health domain [4].

The architecture that was employed made use of the deepseek-r1-distill-qwen-32b [12], qwen2.5-32b [23], and llama3.3-70b [8] models within both the router and specialist agents. Each agent was assigned given prompts in a bid to push its response, with the router making use of a set of prompts in a way to suitably

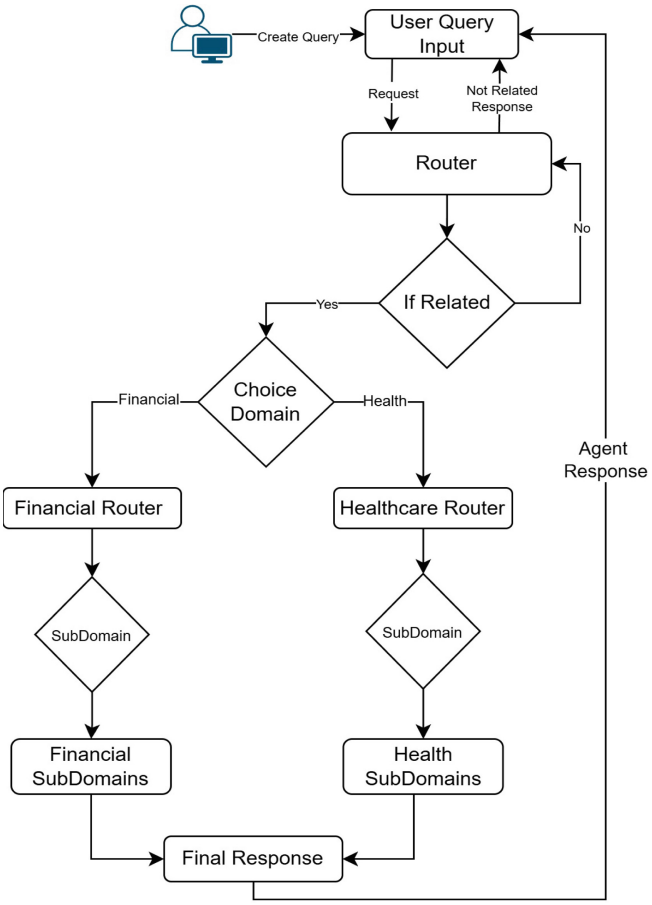

Fig. 1. General Architecture.

understand the intent of queries and direct them towards the most appropriate agent.

The following sections elaborate on the datasets that were used, the complete structure of the architecture, and the prompts used.

5 Experimental Protocol

5.1 Dataset

The choice of the datasets [13], on financial topics, and [4], on the health domain, is explained by the specialized and distinct character of the financial and health domains, with their specific terminologies, question patterns, and contexts. The financial set is applied in testing comprehension and handling of technical topics such as investments, market analysis, and bank products, and the health set

examines the performance of the model to deal with explicit and sensitive inquiry related to medical issues, treatment, and practices in health care.

The financial dataset includes 50 questions, each with two possible answers–one being the "ground truth" and the other from the model that generated the data. Due to the small size, the questions were manually grouped into three broad categories: "Investments", "Financial Planning", and "Others". For the health dataset, which originally had 16,407 questions across several subcategories, a sample of 50 was selected. Only the two most representative groups were kept: "Information" (4,535 questions) and "Symptoms" (2,748 questions), with the rest falling under "Others". Table 2 shows the original distribution, while Table 3 presents the final selection.

Table 2. Subcategories and number of questions present in the health dataset, as well as their percentages in relation to the whole

Category	Quantity	Percentage
support groups	1	0.0001
complications	46	0.0028
stages	77	0.0047
prevention	210	0.0128
considerations	235	0.0143
susceptibility	324	0.0197
outlook	361	0.0220
research	395	0.0241
exams and tests	653	0.0398
causes	727	0.0443
genetic changes	1087	0.0663
frequency	1120	0.0683
inheritance	1446	0.0881
treatment	2442	0.1488
symptoms	2748	0.1675
information	4535	0.2764

To select the questions, percentages were used in relation to the 50 questions to be selected, where each category had at least one sample. With this procedure, 49 samples were selected, and finally, an extra sample was chosen for "Symptoms". After this process, we obtain the following distribution presented in Table 3:

Table 3. Number of questions per subcategory in each category

Finance	Quant.	Health	Quant.
Investments	15	Information	15
Financial Planning	13	Symptoms	09
Others	22	Other	26

5.2 Detailed Architecture and Module Configuration

Figure 2 is an expansion of Fig. 1 and seeks to present the architecture of the experiment carried out, detailing the flow of information from user input to response generation, expanding on how the structure of the Subject Agents is developed given the datasets described in Sect. 5.1. In the following subsections, the model employed and the prompts configured for each module will be described.

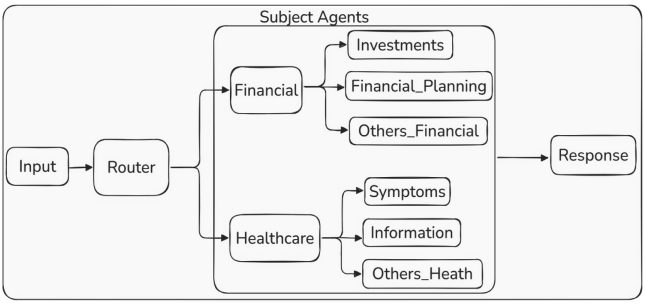

Fig. 2. Detailed Architecture.

Used Models. The architecture employed three state-of-the-art large language models: *deepseek-r1-distill-qwen-32b* [12], *qwen2.5-32b* [23] and *llama3.3-70b* [8] models for both the **router** and the **specialized agents**. These models were selected based on a pragmatic balance of performance, scalability, domain versatility, length of context window, and capacity to handle detailed instructions. These criteria are aligned with contemporary best practices in academic and industrial deployments.

– **deepseek-r1-distill-qwen-32b**: Computationally efficient distilled model with a good trade-off between cost and performance. The distillation here is aimed at retaining response quality with minimal computation, making this model appropriate for applications requiring real-time efficiency [12].

- **qwen2.5-32b**: Powerful contextual understanding model with a longer context length of 128,000 tokens. This makes it highly capable of handling lengthy texts, maintaining coherence and accuracy across a broad variety of domains [23].
- **llama3.3-70b**: Meta AI's advanced model is designed to be highly multilingual, supporting languages such as English, German, French, Italian, Portuguese, Hindi, Spanish, and Thai. It has a context length of 128,000 tokens, allowing for efficient processing of large volumes of data. In addition, its instruction-tuned version is optimized for interactive dialogues and detailed instructions. [8].

Used Prompts. The *prompts* were configured for each component of the architecture, aiming to optimize the interpretation and processing of queries within their areas of activity. The detailed description of the *prompts* used in each module is in appendices A to J. The structuring followed the principles of the CO-STAR framework, which defines five guidelines: provide basic information about the task, define the task to be performed, specify the writing style, determine the response attitude, identify the target audience, and indicate the expected format. The *chain-of-thought* techniques, which add logical steps to the *prompt*, and *few-shot learning*, which uses examples to improve LLM learning, were incorporated.

- **Router**

The routing module relies on a prompt that scans each user request for keywords, context, and implied meaning, paying special attention to cross-domain terms such as "insurance", to identify the primary intent and assign one of two labels, FINANCIAL or HEALTHCARE, before forwarding the query to the corresponding agent; when the input is ambiguous, the label reflects the most dominant subject, and if the request clearly falls outside both domains it is tagged Not Related.

- **Domain Agents**

Once the Router types the query, it is passed on to one of the domain's Agents: *Financial* or *Healthcare*. The agents are domain experts and are tasked with typing the query again into still more specific subdomains so that it can be dealt with even more precisely and accurately. Every Agent uses a *prompt* that instructs it to analyze the query and classify it into one of three categories. For the instance of the *Financial* agent, these categories are:

INVESTMENTS: For investment questions, i.e., stocks, bonds, mutual funds, and portfolio management.

FINANCIAL PLANNING: For questions concerning personal or business financial planning, such as budgeting, saving, and retirement.

OTHERS FINANCIAL: For other questions regarding finances that do not fit the above groups, i.e., general, account management, and unclear.

The *Healthcare* agent places questions in the following groups:

SYMPTOMS: For medical symptom questions, i.e., identification, description, and discussion of an illness.

INFORMATION: For questions asking about general or specific medical information, e.g., treatments, medications, and medical advice unconnected with symptoms.

OTHERS HEALTH: For other than the above requests, such as general inquiries on health, health systems, and ambiguous requests.

The *prompts* also point to the importance of context analysis, keyword detection, and sensitivity to nuances in order to ensure proper classification. They also provide examples of analysis and indicate the structure of the anticipated output.

– **Specialized Agents**

Once classified by the Domain Agents, the query is referred to a Specialized Agent, who responds with the final answer to the user. A Specialized Agent specializes in a specific subcategory of their domain, promoting high specialization in the answers.

Specialized Agents *prompts* are designed in a manner that every one of them turns out to be an expert in his field, providing detailed, personalized, and informative answers. Sample questions and answers are also provided to indicate the nature of the expected interaction.

5.3 Methodology

The process of obtaining the responses was divided into two general phases: orchestration and agent response.

In the orchestration stage, a cascade routing system with LLMs was implemented. The system consists of three prompts that are directed towards classifying and routing the user's question to the concerned agent. The system consists of two routing phases, the router prompt being the first phase and the health and financial prompts being the second phase.

– **Router prompt**: This is the initial level of the cascade and is designed to ascertain the general subject matter of the user's question, classifying it into one of two general categories: finance or health-related issues (Appendix A).
– **Financial and health prompts**: After the general topic has been decided by the router prompt, the second level of routing is initiated. These *prompts* have specific conditions to decide the subtopic within the selected subject and, from it, to choose one of the three agents per subject to answer the question (see Appendices B and C).

In the agent response stage, defined as the final stage of the process, specific prompts were developed for each agent. These prompts contain detailed information about the agent's topic of action, as well as clear instructions on how to formulate the responses.

To evaluate and compare the responses generated by the cascade system, an additional structure with a single comprehensive prompt was implemented. This prompt addresses all the topics defined in the routing stage and includes the corresponding response instructions.

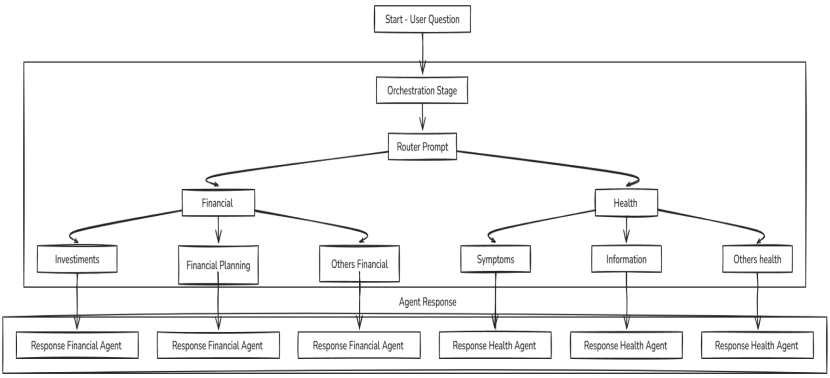

Fig. 3. Waterfall system workflow

6 Results

The findings showed that the organization and presentation of the answers have a great influence on their effectiveness and usability. The contrast between a one-shot approach and another executed by an expert agent shows contrasts in organizational features and detail level and adaptation to the user profile. Where the question is "How do I open a savings account abroad?", the one prompt generated response of the architecture presents information in brief and to the point. It refers, for example, to the need to comply with bank conditions, research foreign institutions and consider factors such as charges and taxations. Such a response, on account of its objectivity, is best suited to users who require an immediate summary of the procedure.

In contrast, the professional agent organizes his answer in a step-by-step fashion, dividing the process into numbered steps from choosing the institution and collecting the documents needed to taking costs, exchange rates and tax implications into account. This step-by-step format explores subtleties and details that can be determinative for those who need detailed and extensive guidance. It should be noted here that there exist studies that illustrate the impact of the length of the prompt on the depth and quality of the response citeprobst2024.

The same was observed in the question "Where to find support for people with Alcohol Use and Older Adults?". While the brief answer mentions important resources such as government institutes and support centers, the expert answer

expands the choices, including specific meetings for the elderly, specialized treatment programs and professional and support group referrals. This makes the answer stronger, anticipating possible doubts and offering alternative means for proper care.

In general, generalist responses are all about efficiency, and they are most appropriately used to get the subject on the table and easy to comprehend immediately. Specialized responses, with their structured format and information-rich depth, facilitate learning and assist in making more informed decisions with less follow-up consultation required.

6.1 Token Distribution by Category

The token distribution comparison by category leads to inferences of differences where the models generate responses in the single prompt strategy and multi-prompt strategy. Figure 4 plots the token distribution for the case of the single prompt, while Fig. 5 describes the result when it is in the multi-prompt.

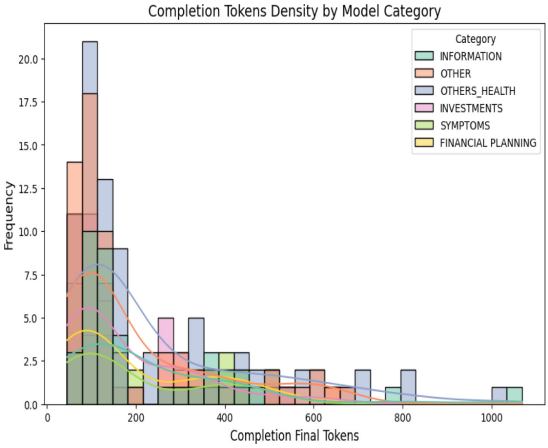

Fig. 4. Distribution of tokens by category in *single prompt*.

The results show that in the single-prompt setup, token distribution is more concentrated, with little variation across categories. This suggests that a single model struggles to adapt to different query complexities, leading to more uniform response lengths. Categories like OTHERS HEALTH, INFORMATION, and OTHER often get shorter replies, while INVESTMENTS and FINANCIAL PLANNING show slightly more variation, though still within narrow limits.

In contrast, the multi-prompt setup shows more variation in token use across categories, as the specialized agents can better match the complexity of each query. INVESTMENTS, for instance, has a wider token spread, pointing to the need for more detailed answers. FINANCIAL PLANNING and OTHERS

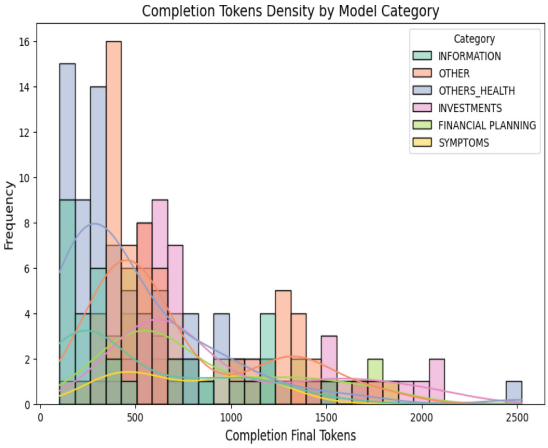

Fig. 5. Distribution of tokens by category in *multi-prompt*.

HEALTH also show broader distributions, indicating more efficient token use when extra context is needed.

The main difference between the two scenarios lies in token flexibility in assignment. There is no segmentation in the single prompt, where less context-adapted responses result, but in the multi-prompt, segmentation of categorization allows the system to maximize response generation to make it more where necessary deep and yet as concise for direct questions.

The gap between the distributions is also statistically confirmed by the ANOVA test [22]. As depicted in Table 4.

Table 4. ANOVA - tokens per category

Cases	Sum of Squares	df	Mean Square	F	p
category	$5.449 \times 10^{+6}$	5	$1.090 \times 10^{+6}$	4.542	< 0.001
Residuals	$1.595 \times 10^{+8}$	665	239901.249	–	–

Note: Type III Sum of Squares

These findings show that the multi-prompt approach optimizes the distribution of tokens and the contextualization of responses, making content generation more efficient and adapted to the specific needs of each category.

6.2 Token Distribution by Model

The distribution of tokens per model allows us to evaluate how different approaches impact resource allocation in response generation. Figure 6 presents the distribution in the single prompt, while Fig. 7 displays the results in the multi-prompt.

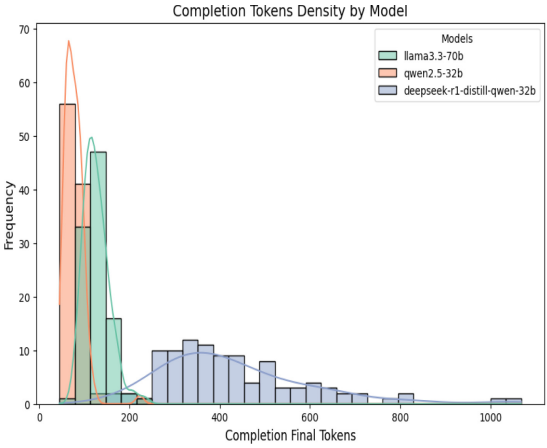

Fig. 6. Distribution of tokens by category in *single prompt*.

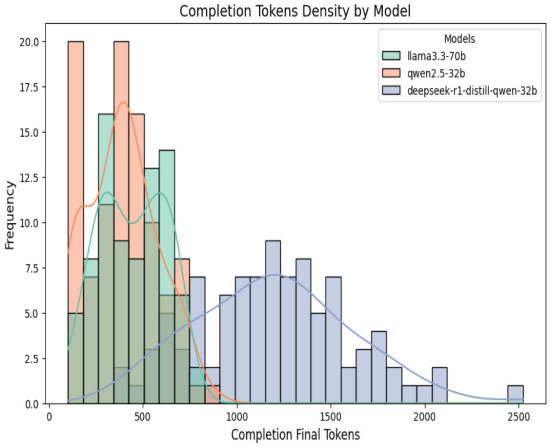

Fig. 7. Distribution of tokens by category in *multi-prompt*.

The results show that in the single-prompt setup, token distribution is more limited, with most responses falling into smaller ranges. qwen2.5-32b tends to give short replies, while llama3.3-70b has a slightly broader but still narrow spread. deepseek-r1-distill-qwen-32b produces longer answers but stays within a moderate range, suggesting the single-prompt method limits how tokens vary across models.

In the multi-prompt scenario, a significantly greater dispersion is observed, especially for deepseek-r1-distill-qwen-32b, which presents a wider distribution and a higher use of tokens in some queries. This behavior suggests that the model adjusts its response according to the complexity of the task. qwen2.5-32b, although still biased towards short responses, exhibits a greater variation, sug-

gesting that, in certain scenarios, it generates more detailed responses. llama3.3-70b maintains an intermediate distribution, adapting better to the context of the questions.

The main difference between the two scenarios is in the adaptability of the models. In the single prompt scenario, the lack of segmentation restricts the variation in the size of the answers, allocating tokens more homogeneously. In the multi-prompt scenario, the system takes advantage of the specialization of the agents to adjust the number of tokens more dynamically, allowing each model to optimize its answers according to the complexity of the query.

As with the distributions presented in the previous section, the difference between these distributions is statistically proven via the ANOVA test [22], presented in Table 5.

Table 5. ANOVA - tokens per model

Cases	Sum of Squares	df	Mean Square	F	p
model	$1.237 \times 10^{+8}$	5	$2.473 \times 10^{+7}$	397.996	< 0.001
Residuals	$4.132 \times 10^{+7}$	665	62140.919	–	–

6.3 Implications for the Proposed Architecture

The results show that the multi-prompt approach produces more adaptive and effective answers compared to a single prompt. In the single prompt, there is uniform token distribution, limiting the responses' adequacy to the question's complexity. In the multi-prompt, there is greater variance in the token distribution, allowing more complex responses for complex questions and greater conciseness in simple questions.

The specialization of the models was also greater in the multi-prompt, where every agent optimized its token distribution in light of task demands. Such models as deepseek-r1-distill-qwen-32b were more adaptable in generating longer output, while qwen2.5-32b produced a shorter pattern. The findings hold that modularity enhances the efficacy of a system by facilitating more strategic deployment of the models.

Aside from response quality, modular architecture facilitates computation optimization, spreading queries across expert agents and avoiding resource wastage. Thus, the multi-prompt strategy improves the accuracy, scalability, and efficiency of the system, highlighting the need for smart routing and model specialization for advanced LLM applications.

7 Conclusion

This paper presented a chatbot architecture from LLMs, routing user inquiries through a routing module to targeted agents. The solution proposed seeks to

improve response accuracy and contextualization and reduce the rate of hallucinations that accompany the independent use of a single model.

The architecture was evaluated through a case study using data from the health and finance domains, evidencing satisfactory results in the classification and handling of queries. The use of a modular structure, combined with the division into specialized agents, demonstrated greater efficiency in processing questions, in addition to offering flexibility for adaptations to new scenarios.

In addition, the proposed approach is effective in dealing with multiple domains, minimizing the need for extensive volumes of data for training or the creation of overly complex prompts. Modularity allowed segmentation into specific and objective prompts, each focused on a distinct task, significantly reducing the risk of confusion regarding the intentions of the queries. Thus, the modular structure with specialized agents proved to be a promising solution.

Among the limitations identified, the limited amount of data available for testing stands out, as well as the need for improvements in the prompts used. For future work, it is recommended to expand the tests with the modular architecture, include more data for validation, and improve the prompt engineering, in addition to optimizing the models used.

The proposed architecture, although initially applied to the health and finance domains, is designed to be extended. As future work, we propose incorporating other domains such as law and education, as well as expanding the dataset to evaluate the scalability of the solution, and integrating more robust quantitative metrics, such as accuracy, F1-score, and hallucination rate, to provide a more comprehensive evaluation of the system's performance.

References

1. Alabed, A., Javornik, A., Gregory-Smith, D.: AI anthropomorphism and its effect on users' self-congruence and self–AI integration: a theoretical framework and research agenda. Technol. Forecast. Soc. Change **182**, 121786 (2022). https://doi.org/10.1016/J.TECHFORE.2022.121786
2. Bai, Y., et al.: Pistis-RAG: Enhancing retrieval-augmented generation with human feedback. arXiv preprint arXiv:2407.00072 (2024)
3. Bakkouri, B.E., Raki, S., Belgnaoui, T.: The role of chatbots in enhancing customer experience: literature review. In: Procedia Computer Science. vol. 203, pp. 432–437. Elsevier B.V. (2022). https://doi.org/10.1016/j.procs.2022.07.057
4. Ben Abacha, A., Demner-Fushman, D.: A question-entailment approach to question answering. BMC Bioinform. **20**(1) (2019). https://doi.org/10.1186/s12859-019-3119-4
5. Chaurasia, A., Parashar, B., Kautish, S.: Artificial intelligence and automation for industry 4.0. In: Computational Intelligence for Modern Business Systems: Emerging Applications and Strategies, pp. 357–373. Springer (2023)
6. Chen, J.S., Le, T.T.Y., Florence, D.: Usability and responsiveness of artificial intelligence chatbot on online customer experience in e-retailing. Int. J. Retail Distrib. Manag. **49**, 1512–1531 (2021). https://doi.org/10.1108/IJRDM-08-2020-0312
7. Ding, D., et al.: Hybrid LLM: Cost-efficient and quality-aware query routing. arXiv preprint arXiv:2404.14618 (2024)

8. Dubey, A., et al.: The Llama 3 herd of models (2024). https://arxiv.org/abs/2407.21783
9. van Dun, C., Moder, L., Kratsch, W., Röglinger, M.: Processgan: Supporting the creation of business process improvement ideas through generative machine learning. Decis. Support Syst. **165**, 113880 (2023). https://doi.org/10.1016/j.dss.2022.113880, https://linkinghub.elsevier.com/retrieve/pii/S0167923622001518
10. Feng, T., Shen, Y., You, J.: GraphRouter: A graph-based router for LLM selections. arXiv preprint arXiv:2410.03834 (2024)
11. Fui-Hoon Nah, F., Zheng, R., Cai, J., Siau, K., Chen, L.: Generative AI and ChatGPT: Applications, challenges, and AI-human collaboration (2023)
12. Guo, D., et al.: DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. arXiv preprint arXiv:2501.12948 (2025)
13. Acessado em 26 de novembro de 2024
14. Li, C.Y., Fang, Y.H., Chiang, Y.H.: Can AI chatbots help retain customers? An integrative perspective using affordance theory and service-domain logic. Technol. Forecast. Soc. Change **197**, 122921 (2023). https://doi.org/10.1016/J.TECHFORE.2023.122921
15. Ong, I., et al.: RouteLLM: Learning to route LLMs with preference data. arXiv preprint arXiv:2406.18665 (2024)
16. Putha, S.: Ai-driven personalization in e-commerce: enhancing customer experience and sales through advanced data analytics. J. Bioinform. Artif. Intell. **1**(1), 225–271 (2021)
17. Roychowdhury, S., Krema, M., Mahammad, A., Moore, B., Mukherjee, A., Prakashchandra, P.: ERATTA: Extreme rag for table to answers with large language models. arXiv preprint arXiv:2405.03963 (2024)
18. Shi, L., Kazda, M., Sears, B., Shropshire, N., Puri, R.: Ask-EDA: A design assistant empowered by LLM, hybrid rag and abbreviation de-hallucination. arXiv preprint arXiv:2406.06575 (2024)
19. Sidaoui, K., Jaakkola, M., Burton, J.: AI feel you: customer experience assessment via chatbot interviews. J. Serv. Manag. **31**, 745–766 (2020). https://doi.org/10.1108/JOSM-11-2019-0341
20. Stripelis, D., et al.: PolyRouter: a multi-LLM querying system. arXiv e-prints, pp. arXiv–2408 (2024)
21. Stripelis, D., et al.: TensorOpera router: A multi-model router for efficient LLM inference. arXiv preprint arXiv:2408.12320 (2024)
22. Sthle, L., Wold, S.: Analysis of variance (ANOVA). Chemom. Intell. Lab. Syst. **6**(4), 259–272 (1989)
23. Yang, A., et al.: Qwen2. 5 technical report. arXiv preprint arXiv:2412.15115 (2024)
24. Zhang, Y., Pei, H., Zhen, S., Li, Q., Liang, F.: Chat generative pre-trained transformer (ChatGPT) usage in healthcare. Gastroenterol. Endosc. **1**(3), 139–143 (2023)