Towards Trustworthy and Efficient Smart Routing for Large Language Models

ROULE Jule[1], *ILHE Paul[2], MOUAYAD Mehdi[1], MAZARS Gilles[2], and BARRY Mariam[1]

[1]BNP Paribas
[2]Vector8

1 Introduction

The proliferation of Large Language Models (LLMs) has created a landscape where dozens of models with varying capabilities, costs, and specializations are available for deployment. This abundance presents a critical challenge: how to intelligently select the optimal model for each specific query based on a single, well-defined optimization criterion. Traditional approaches (see (Varangot-Reille, 2025)) often rely on static model selection or expensive flagship models for all tasks, leading to suboptimal resource utilization and unnecessary expenses.

In this paper, we introduce a novel dynamic routing architecture that addresses this challenge through an intelligent and fast system. Our approach leverages model-specific performance datasets to construct comprehensive leaderboards that track metrics across different skill domains. At inference time, we employ gpt-4o-mini for real-time skill extraction, enabling our router to make informed routing decisions based on the extracted skill and a single optimization target.

For each skill domain, we store the correctness and cost metrics, allowing our system to optimize for a single criterion, such as correctness or cost. This focused approach ensures predictable routing behavior while maintaining computational efficiency.

Our dynamic router identifies the model that achieves the highest score for the chosen optimization criterion within that skill domain. This single-objective maximization provides transparency and interpretability, making it easier to validate routing decisions.

The architecture is designed for domain specialization while maintaining the flexibility to accommodate new models without complete system retraining. Performance metrics for new models can be integrated into existing leaderboards, ensuring our routing system remains current as the LLM landscape evolves, as opposed to (Chen, 2024), which needs retraining for any newly added model.

Preliminary evaluations demonstrate that our single-objective routing strategies can achieve significant cost improvements while preserving a similar correctness score.

*corresponding Author: paul.ilhe@vector8.com

2 Proposed Dynamic Routing

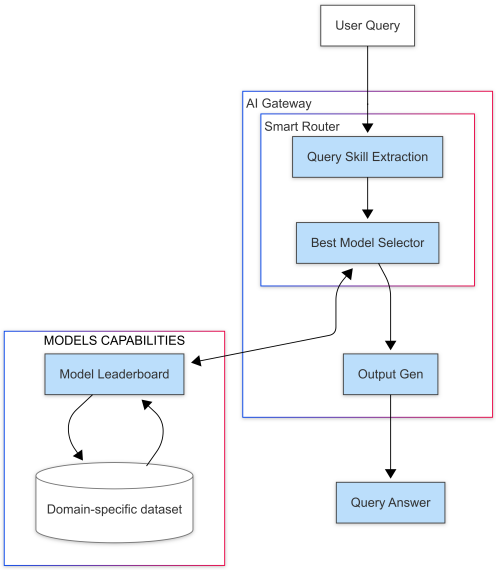

Figure 1: Architecture diagram

The dynamic router consists of four main components:

1. Skill-Aware Query Analysis

Upon receiving a query, the system utilizes a fast language model (gpt-4o-mini in this paper) to extract the specific skill category required for an effective response. This enables the router to match queries with the most suitable model capabilities in real time.

2. Offline Model Evaluation and Leaderboard Construction

Models are evaluated offline using public benchmarks such as `openai_humaneval`, `SimpleQA`, `openai_healthbench`, and `wmt_14`. The evaluation leverages the library `deepeval` to assess the correctness (via a LLM-as-a-judge) and the cost for each skill domain. The results populate a dynamic leaderboard that serves as the foundation for routing decisions.

3. Single-Objective Model Selection

At inference time, the router consults the leaderboard to select the model that maximizes the following optimization criterion for the extracted skill

$$\text{correctness} - \lambda \cdot \text{cost} \ ,$$

where λ is a cost penalty factor. Selection strategies include pure correctness, cost-aware penalties, or other performance measures. This approach provides predictable and interpretable routing behavior.

4. Adaptability and Scalability

The architecture supports ongoing integration of new models: as models are released, their performance is evaluated and metrics are added to the leaderboard without retraining the entire system. This enables multiple applications to leverage the router as a unified service, abstracting away model-specific adaptations and supporting seamless, domain-specialized routing.

3 Results

To evaluate our method, we selected four diverse public datasets: `openai_humaneval`, `SimpleQA`, `openai_healthbench`, and `wmt_14`. These datasets span a range of domains including health, coding, translation, and general knowledge and together comprise more than 900 samples, ensuring a broad representation of real-world query types and difficulty levels.

Our experiments utilized the following models from Mistral AI and XAI: `mistral-medium-2508`, `mistral-large-2411`, `mistral-small-2506`, `grok-3-fast`, `grok-3`, `grok-3-mini`, and `grok-code-fast-1-0825`.

We compared the router's performance across four different criterion with penalty factors λ (0, 5, 15, and 50), benchmarking against an oracle router that always selects the best possible answer for each request category, as well as against the individual models. Figure 2 illustrates this comparison, plotting average score versus average cost for 1 million queries.

While the router's performance remains below the oracle as displayed in Figure 2, the use of penalized routers leads to a substantial improvement (of 1.5%) compared to the best individual model, while reducing cost by a factor of 18. Notably, the router with cost penalty 15 achieves a similar average correctness score as the unpenalized version (53.4% vs. 53.1%), but with a tenfold reduction in cost.

This performance is achieved by selecting less expensive models as reported in Figure 3 where models like `mistral-medium-2508` and `grok-3-mini` tend to be more selected for queries with nearly similar correctness scores.

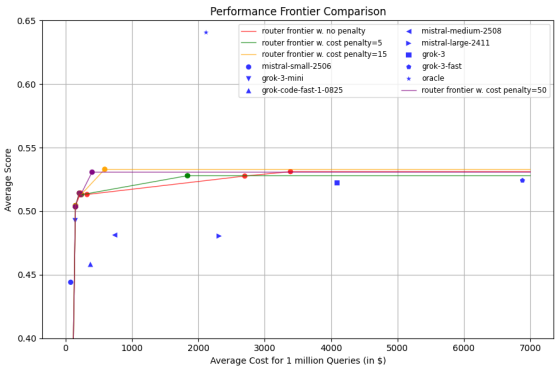

Figure 2: Performance Frontier Comparison

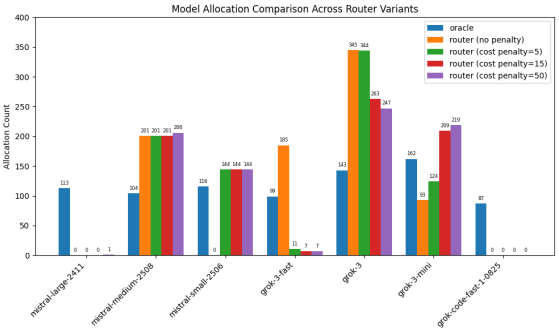

Figure 3: Model allocation comparison

4 Conclusion and future work

Model routing shares similarities with recommendation systems, as both require effective representations of item (or model) capabilities and a mechanism to match them with the specific requirements of incoming queries. In this work, we implemented a basic matching approach, utilizing sample categories from our dataset to encode model capabilities. Despite its simplicity, this method demonstrates strong performance.

For future work, we aim to develop more fine-grained representations of model capabilities, to capture subtle distinctions between models while maintaining the low overhead introduced by classifying queries with `gpt-4o-mini`. This advancement has the potential to further enhance routing accuracy and efficiency in practical applications.

References

Chen, S. e. a. (2024). Routerdc: Query-based router by dual contrastive learning for assembling large language models. *Advances in Neural Information Processing Systems*, *37*, 66305–66328.

Varangot-Reille, C. e. a. (2025). Doing more with less–implementing routing strategies in large language model-based systems: An extended survey.