University of Reading

Department of Computer Science

Routers for LLM: A Framework for Model Selection and Tool Invocation

Ruben J. Lopes

Supervisor: Dr. Xiaomin Chen

A report submitted in partial fulfilment of the requirements of
the University of Reading for the degree of
Bachelor of Science in *Computer Science*

May 6, 2025

Declaration

I, Ruben J. Lopes of the Department of Computer Science, University of Reading, confirm that this is my own work and figures, tables, equations, code snippets, artworks, and illustrations in this report are original and have not been taken from any other person's work, except where the works of others have been explicitly acknowledged, quoted, and referenced. I understand that if failing to do so will be considered a case of plagiarism. Plagiarism is a form of academic misconduct and will be penalised accordingly.

I give consent to a copy of my report being shared with future students as an exemplar.

I give consent for my work to be made available more widely to members of UoR and public with interest in teaching, learning and research.

Ruben J. Lopes
May 6, 2025

Abstract

In the current landscape of large language models, users are confronted with a plethora of models and tools each offering a unique blend of specialisation and generality. This project proposes the development of a dynamic middleware "router" designed to automatically assign user queries to the most appropriate model or tool within a multi-agent system. By using zero-shot Natural Language Inference models, the router will evaluate incoming prompts against criteria such as task specificity and computational efficiency, and *route* the prompt to the most effective model and/or allow specific tools relevant that the model could use.

The proposed framework is underpinned by three core routing mechanisms:

- Firstly, it will direct queries to cost effective yet sufficiently capable models, a concept that builds on existing work in semantic routing Ong et al. (2025).
- Secondly, it incorporates a tool routing system that automatic invocation of specialised functions, thus streamlining user interaction and reducing inefficiencies currently inherent in systems like OpenAI's and Open Web UI. Furthermore this could also reduce inefficiencies in the recent reasoning models addressing the observed dichotomy between underthinking with complex prompts and overthinking with simpler queries when reasoning is manually toggled which can be costly and could cause hallucination.
- Thirdly, while the primary focus remains on model and tool routing, this work will preliminarily explore the potential application of the routing architecture as a security mechanism. Initial investigations will examine the theoretical feasibility of leveraging the router's natural language understanding capabilities to identify adversarial prompts. This includes a preliminary assessment of detection capabilities for prompt engineering attempts, potential jailbreaking patterns, and anomalous tool usage requests. However, given the rapidly evolving nature of LLM security threats and the complexity of implementing robust safeguards, comprehensive security features remain outside the core scope of this research. This aspect represents a promising direction for future work, particularly as the field of LLM security continues to mature.

By integrating these mechanisms, the research aims to pioneer a more efficient, modular, and secure distributed AI architecture. This architecture not only optimises resource allocation but also reinforces system integrity against emerging adversarial threats, thereby contributing novel insights into the development of next generation LLM deployment strategies.

Acknowledgements

An acknowledgements section is optional. You may like to acknowledge the support and help of your supervisor(s), friends, or any other person(s), department(s), institute(s), etc. If you have been provided specific facility from department/school acknowledged so.

Contents

List of Figures

List of Tables

List of Abbreviations

SMPCS	School of Mathematical, Physical and Computational Sciences
AI	Artificial Intelligence
GPT	Generative Pre-trained Transformer
MAS	Multi-agent systems
MCP	Multimodal Context Processing
MoE	Mixture of Experts
NLI	Natural Language Inference
NLP	Natural Language Processing

Chapter 1

Introduction

In the past few years the landscape of large language models has expanded dramatically, with many domain specific as well as general purpose agents emerging across domains such as healthcare (like Med PaLM 2 and BioGPT), coding (like CodeLlama and GitHub Copilot), and research (like Claude Opus and GPT 4o). Organisations that provide inference as a service now face complex trade offs between cost, latency, and capability; for example, GPT 4.5 can cost up to $75 per million tokens compared with just $0.15 for gemini 2.5 flash[1][2]. Although these models could be vastly different in terms of capability, the problem organisations face is determining *when* to deploy premium models versus more cost effective alternatives for a given task / prompt. This suggests a need for intelligent routing systems that can analyse incoming prompts and direct them to the most appropriate model based on task complexity, required capabilities, and cost considerations.

Inspired by lower level (transformer embedded) "router" such as the one employed by mistral for their Mixtral (MoE) model the goal of this was to allow for a more distributed, higher level prompt based routing between a verity of models with varying levels of cost and complexity.

1.1 Problem statement

The proliferation of large language models has created a complex ecosystem where selecting the optimal model for a given task has become increasingly challenging.

Existing multi agent routing systems reveal several shortcomings. First, many current routers rely on **manual configuration**. For example, Both Open AI's as well as Open WebUI's chat interface require explicitly toggling of tools/selection of agents from users, and listing models to skip, on a per chat basis. Second, LLM based routers can suffer from reasoning **inefficiencies**. Recent studies identify *"underthinking"* (prematurely abandoning good reasoning paths) and *"overthinking"* (generating excessive, unnecessary steps) in modern LLMs. For instance, in a study Wang et al. (2025), the authors find that top reasoning models often switch thoughts too quickly – an *"underthinking"* effect that hurts accuracy. Conversely, Kumar *et al.* demonstrate how even simple queries can be made to "overthink" (spending many tokens on irrelevant chains of thought) without improving answers Kumar et al. (2025). *Overthinking* is particularly problematic in the context of function calling, where excessive reasoning can lead to unnecessary API calls and increased costs or worse, hallucinations. This is especially relevant for models like GPT 4o and Claude 3 Opus, which are designed to handle

[1] https://sanand0.github.io/llmpricing/

[2] https://artificialanalysis.ai/

complex reasoning tasks but can rack up significant costs if not used sparingly. The recent introduction of function calling in LLMs has further complicated this landscape, as users must now navigate a myriad of specialised tools and functions. This complexity can lead to inefficient routing decisions, where users may inadvertently select more expensive or less suitable models for their tasks. Finally, prompt interpretation remains imperfect: ambiguous or poorly phrased queries may be misrouted or require multiple LLM calls to resolve intent, leading to inefficiency.

Organisations and users face several key problems:

1. **Cost Efficiency Trade offs**: High capability models like GPT 4o and Claude 3 Opus provide powerful capabilities but at significantly higher costs than simpler models. Without intelligent routing, organisations and users may unnecessarily infer to expensive models for tasks that could be adequately handled by more cost effective alternatives.

2. **Selection Complexity**: With the dawn of function calling and Multimodal Context Processing (MCP), most chat systems offer numerous specialised tools and functions, but determining which tools are appropriate for a given query often requires manual specification by users or developers.

3. **Computational Resource Allocation**: Indiscriminate routing of all queries to high performance models can lead to inefficient resource allocation, increased latency, and higher operational costs for LLM providers and users.

1.2 Research Objectives

The premise of this research is to investigate whether pre existing Natural Language Inference models such as Facebook's bart-large-mnli could be used as drop in replacements to perform automated model selection and tool selection and potentially even using it as a security mechanism to detect adversarial prompts. Furthermore, we will examine the effectiveness of finetuning existing NLI models with specialised datasets designed for routing tasks.

The specific research objectives include:

- Creating a LLM Router library that can be deploy to existing systems with ease.

- Experimenting with Pretrained NLI models such as bart-large-mnli for both tool routing and model selection.

- Evaluating and assessing the accuracy the effectiveness using a set of prompts.

- Incorporate it with an existing Chatbot UI platform such as OpenWebUI.

Natural Language Inference (NLI) is a subfield of Natural Language Processing (NLP) that focuses on determining the relationship between pairs of sentences. This is essential for what we are trying to achieve, as it allows us to understand the semantic relationship between a given prompt and the capabilities of different models or tools using its descriptions. By leveraging NLI techniques, we can create a more efficient and effective routing system that can automatically select the most appropriate model or tool for a given task.

Chapter 2

Literature Review

2.1 Large Language Models: Current Landscape

Large scale LLMs continue to grow in parameter count and capability, intensifying the trade off between performance and computational cost. Models such as OpenAI's GPT 4 and Google's Gemini 2.5 Pro deliver top tier results, but at significantly higher inference costs often 400 to 600 times more than comparable alternatives [1]. With many state of the art models being closed source (only accessible through an API), a new wave of open weight and open source models has emerged. These models make it easier for individuals and companies to self host, potentially lowering operational costs. For organisations offering inference as a service, open models are particularly advantageous not only for cost efficiency, but also for addressing privacy and security concerns associated with sending user prompts to third party providers.

2.2 Multi-Agent Systems and Distributed AI Architecture

Multi-agent systems (MAS) have been a subject of research and development since the 1980s. While traditional MAS research established fundamental principles by using agent communication protocols such as KQML and FIPA-ACL, the emergence of Large Language Models has transformed how these systems operate in practice.

In December 2023, Mistral AI introduced Mixtral 8x7B, a model that employs a Sparse Mixture of Experts (MoE) architecture suggesting a promising approach which only activates a subset of a large model's "experts" per query (Fu et al., 2025). This gave them the edge over other models such as Llama 2 70B on most benchmarks where Inference was 6 times faster and even matches or outperforms GPT 3.5 on most benchmarks (Hu et al., 2024). While Mixtral applies routing at the model architecture level rather than through a separate system level orchestration, it demonstrated the potential for such a middle layer.

2.3 Semantic Routing Mechanisms

Several recent projects provide router-like middleware to manage multi model access. OpenRouter.ai provides a unified API that hides model providers behind a single endpoint, dynamically routing requests across providers to optimise cost and availability. On the open source side, RouteLLM formalises LLM routing as a machine learning problem, with results showing "cost reductions of over 85% on MT Bench while still achieving 95% of GPT 4's

[1] https://help.openai.com/en/articles/7127956-how-much-does-gpt-4-cost

performance" [2]. Another routing mechanism, Router Bench, shows promise with over 405,000 inference outcomes from representative LLMs (Hu et al., 2024).

On the tool routing side, landmark papers like Toolformer (Schick et al., 2023) demonstrate how LLMs can learn to invoke tools. At the interface level, OpenAI's Function Calling and "built in tools" features have begun to infer tool usage directly from user prompts.

2.4 Routing Approaches

In evaluating alternatives, several decision making mechanisms currently used by LLM services are:

- **Rule based routing:** This relies on predefined heuristic rules or configuration files (**?**). Each routing decision is directly traceable to an explicit rule (**?**). However, it often lacks contextual understanding.
- **Prompt based routing:** This involves invoking a language model with a crafted system prompt. The model's response is passed to the relevant tool or agent.
- **Similarity Clustering based Routing:** This method leverages unsupervised clustering algorithms to group historical user queries (**?**).
- **NLI based (zero shot) routing:** This employs a pre trained Natural Language Inference model for zero shot intent classification.

2.5 Research Gap Analysis

As highlighted previously, multi agent routing has been successfully implemented both as closed source (OpenRouter.ai) and in open source libraries such as RouteLLM. However, there remains significant opportunity for innovation in this space.

[2] https://lmsys.org/blog/2024-07-01-routellm

Chapter 3

Methodology

Chapter 4

Results

Chapter 5

Discussion and Analysis

Chapter 6

Conclusions and Future Work

Chapter 7

Reflection

References

Fu, Y., Jiang, Y., Huang, Y., Nie, P., Lu, Z., Xue, L., He, C., Sit, M.-K., Xue, J., Dong, L., Miao, Z., Zou, K., Ponti, E. and Mai, L. (2025), 'Moe-cap: Benchmarking cost, accuracy and performance of sparse mixture-of-experts systems'.
URL: *https://arxiv.org/abs/2412.07067*

Hu, Q. J., Bieker, J., Li, X., Jiang, N., Keigwin, B., Ranganath, G., Keutzer, K. and Upadhyay, S. K. (2024), 'Routerbench: A benchmark for multi-llm routing system'.
URL: *https://arxiv.org/abs/2403.12031*

Kumar, A., Roh, J., Naseh, A., Karpinska, M., Iyyer, M., Houmansadr, A. and Bagdasarian, E. (2025), 'Overthink: Slowdown attacks on reasoning llms'.
URL: *https://arxiv.org/abs/2502.02542*

Ong, I., Almahairi, A., Wu, V., Chiang, W.-L., Wu, T., Gonzalez, J. E., Kadous, M. W. and Stoica, I. (2025), 'Routellm: Learning to route llms with preference data'.
URL: *https://arxiv.org/abs/2406.18665*

Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Zettlemoyer, L., Cancedda, N. and Scialom, T. (2023), 'Toolformer: Language models can teach themselves to use tools'.
URL: *https://arxiv.org/abs/2302.04761*

Wang, Y., Liu, Q., Xu, J., Liang, T., Chen, X., He, Z., Song, L., Yu, D., Li, J., Zhang, Z., Wang, R., Tu, Z., Mi, H. and Yu, D. (2025), 'Thoughts are all over the place: On the underthinking of o1-like llms'.
URL: *https://arxiv.org/abs/2501.18585*

Appendix A

An Appendix Chapter (Optional)

Appendix B

An Appendix Chapter (Optional)