Intelligent Planner: A Dynamic Query Routing System Across Large Language Models

Chidaksh Ravuru chidaksh@cs.unc.edu
Pavan Akkineni akkineni@cs.unc.edu
Saksham Khajuria sakk@cs.unc.edu
Ruthvika Kosuri rkosuri@cs.unc.edu

Abstract

The availability of multiple Large Language Models and their competencies in various domains has brought an idea of leveraging these domain-specific strengths of individual models to get the best response through an agentic setup.[6] Intelligent Planner comes up with a dynamic query routing system by classifying these queries and directing them to the model that has shown best results for that category based on our experiments. Evaluation of each model's response on dataset generated using SQuAD, MMLU, PIQA and GSM8K was used to fine-tune a classification component using Llama. The summarization of previous query responses in a single chat-based environment is also achieved by appending the query after it has been classified to a domain type with the summary of previous responses in order to provide a contextual memory.

1 Introduction

The dramatic evolution of large language models (LLMs) has opened unprecedented possibilities for natural language processing applications but revealed a inherent flaw: single models differ in performance across different domains and task types. While some models excel very well in mathematical reasoning, others excel in creative writing or commonsense reasoning. This variability becomes particularly bothersome when computational resources are scarce as running a number of large models in parallel is often infeasible.

This research addresses these challenges by developing the Intelligent Planner, a novel query routing system that dynamically routes user queries to specialized language models automatically based on classification. By leveraging the unique strengths of different models through intelligent routing, our system performs better than any single-model solution while remaining within practical computational limits.

Our model integrates four quantized language models (Mistral, Llama, Qwen, and Gemma) into a unified architecture with an LlaMA 3.1-8B-based intelligent query classifier. The classifier categorizes incoming queries into eight domains: Natural Language Understanding, Reading Comprehension, Cognitive Reasoning, Natural Sciences, Common Sense, Social Sciences, Mathematics, and a generic Fallback category. Experimental evaluation verifies that our system always beats individual models for various kinds of queries, confirming the value of intelligent routing systems for streamlining language model interactions.

2 Background

The emergence of different large language models has revealed distinct performance profiles across different domains. While models like GPT-4 and Claude are general-purpose systems with broad capabilities, open-source alternatives like Llama, Mistral, and Qwen are specialized in specific types of tasks. Such diversity offers the potential for optimization through intelligent routing systems.

Traditional query routing methods take a cue from ensemble learning but will incur multiple models running simultaneously and cause computational issues. Quantization models address such limitations by restraining memory consumption without compromising performance. Our model fuses classification, routing, and memory management to function under real-world constraints.

Benchmark testing reveals drastic performance contrasts: Qwen performs well on mathematical reasoning (GSM8K), while Mistral excels at social sciences tasks (MMLU). These patterns motivated the development of a routing scheme that opportunistically leverages each model's strengths rather than relying on one solution.

Good memory management is a balance between maintaining context and being computationally efficient. Our double-memory approach maintains recent interactions in the foreground automatically while keeping explicit access to previous queries readily available, sustaining conversational coherence under resource scarcity.

Comprehensive assessment requires multiple metrics. Our model combines syntactic measures (F1-score, BLEU, RougeL), semantic measures (embedding similarity), and context-sensitive measures (BERTScore) to fully estimate performance by task type.

3 Related Works

The Intelligent Planner leverages growing amounts of research that examine expert model routing, multi-model orchestration, and the foundation research on large language models (LLMs). Various leading projects constitute the conceptual and architectural bases of this project.

GLIDER (Li et al., 2024) introduced a global and local instruction-guided expert router to address the limitations of existing routing mechanisms for expert models. Through combining a token-aware local router and a semantic-aware global router, GLIDER exhibited significant advances on held-in and held-out tasks. The double-scale routing inspired our multi-layered classification and selection mechanism in this project and also endeavors to optimize over a heterogeneous taxonomy of tasks with LLM-delivered task awareness to select the right expert.[1]

HuggingGPT (Shen et al., 2023) presented a language-driven interface positioning LLMs as planners that can plan and execute multi-modal AI processes by orchestrating external expert models using model cards from Hugging Face. The system positions the LLM as a planner, an idea that is mimicked in the Intelligent Planner architecture, where the classifier is positioned as a centralized decision-maker dynamically assigning jobs to domain-specialized models.[4]

Schaeffer (2024) gave a step-by-step tutorial on fine-tuning LLaMA 3.1 models step by step on easily accessible platforms such as Google Colab and Hugging Face. His emphasis on quantized, efficient deployment is highly compatible with the Intelligent Planner's utilization of 4-bit quantized models to trade computational resources for zero performance loss. While Schaeffer's work is implementation-focused, it lends support to the practicability and simplicity of fine-tuning high-performing models in resource-limited settings.[3]

Finally, Xiao and Zhu (2025) give us an underlying background of LLMs such as pretraining, prompting, alignment, and scaling techniques. Their work gives theoretical context to the architectural decisions in our system, particularly on instruction tuning, fine-tuning techniques like Direct Preference Optimization (DPO), and prompt-based adaptation. These underlying concepts informed our classification and routing system, particularly how we proceeded with designing prompts and aligning models.[5]

Together, the efforts confirm the modularity and cooperation of AI systems where a controller (often an LLM) leverages heterogeneous specialist model expertise. The Intelligent Planner extends this structure by integrating routing, classification, context storage, and performance statistics into a lightweight, streamlined system with real-time task allocation and domain-sensitive query answering.

4 Data

4.1 Benchmark Datasets

Our assessment was on four standard benchmark datasets, one chosen to measure different cognitive and domain-specific capabilities:

SQuAD (Stanford Question Answering Dataset): A reading comprehension dataset with over questions posed by crowdworkers to Wikipedia documents. The responses are text spans of the corresponding passages. We applied this dataset to measure reading comprehension and information extraction ability.

MMLU (Massive Multitask Language Understanding): Overall task evaluating knowledge in 57 STEM, humanities, social sciences, and professional domains. The tasks vary from elementary school levels to professional levels, measuring breadth and depth of model knowledge.

PIQA (Physical Interaction Question Answering): A common sense physical reasoning test dataset through 4,000 questions asked on everyday scenarios. Developed to check whether models have a sense of basic physical interactions and object features that are innate to humans.

GSM8K (Grade School Math 8K): Grade school math word problem dataset with 4000 multistep reasoning problems. The problems challenge arithmetic, algebra, and reading comprehension of word problems.

4.2 Synthetic Training Data

We have produced synthetic preference data to train our routing mechanism through:

- Querying all four models (Mistral, Llama, Qwen, Gemma) with questions from each benchmark
- Collecting both the model's answer and internal reasoning process
- Scoring responses using multiple metrics against ground truth answers
- Creating preference pairs based on performance differences

5 Methodology

Our project developed an intelligent routing system to direct user queries to the most appropriate language model. The basic idea was straightforward: various models are good at different things, so querying specialized models results in better answers than querying a single general-purpose model (Figure 3).

For our base, we selected four quantized language models: Mistral, Llama, Qwen, and Gemma. Each model possessed varying strengths for varying types of tasks, and the utilization of 4-bit quantized versions allowed us to cope with the limited GPU memory that we had at our disposal in our environment. This quantization reduced memory usage considerably while having little effect on model performance.

The training process proceeded through three well-defined phases (Figure 1): We first created artificial training data by querying common benchmarks extensively. We used SQuAD, MMLU, PIQA, and GSM8K datasets to collect model responses and chains of reasoning. Each response was graded against ground truth answers with F1 scores and embedding similarity measures. This data collection formed the basis for subsequent training.

In the second phase, we fine-tuned a classification component using meta-llama/Llama-3.1-8B-Instruct. This classifier learned to identify which model would best handle each query type through binary classification between response alternatives. The training was focused on detecting quality differences in model answers.

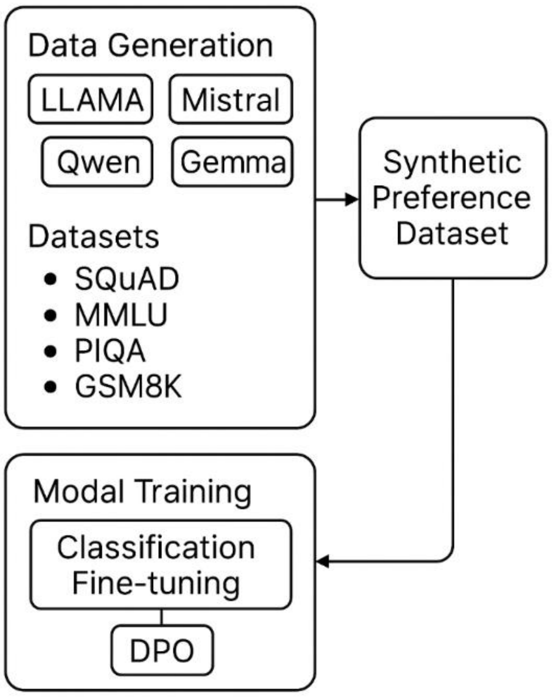


Figure 1: LLM Training Pipeline detailing data generation, classification fine-tuning, and direct preference optimization (DPO)

The third phase used Direct Preference Optimization [2] in the sense that for each question, it compared three objects: the prompt, the accepted model response, and the rejected model response. This technique regularly adapted the system's routing decisions through comparative testing.

Query classification was the result of intense performance analysis. Our evaluation revealed crisp patterns: Llama did Natural Language Understanding the best, Mistral dominated Reading Comprehension and all social science subcategories, Qwen did Mathematics and Cognitive Reasoning the best, and Gemma was our choice for borderline cases. Routing was simple: classify the incoming query, select the best-performing model for the associated category, then serve the answer.

Memory management was essential to maintaining conversational continuity. We maintained a two-tier arrangement: general memory retained the previous five interactions in automatic mode, while specific memory allowed users to recall prior queries by their unique identifiers. The system abstracted conversation history prior to handling new queries, retaining relevant context in store without flooding the processing capacity (Figure 2).

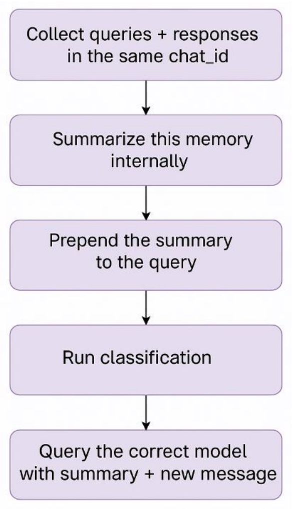

Collect queries + responses in the same chat_id
Summarize this memory internally
Prepend the summary to the query
Run classification
Query the correct model with summary + new message

Figure 2: In-chat Contextual Memory

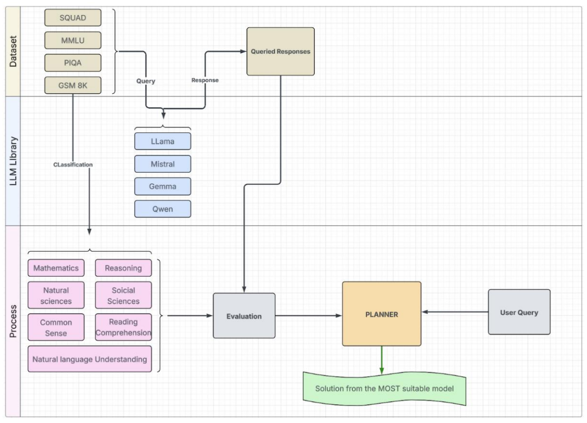
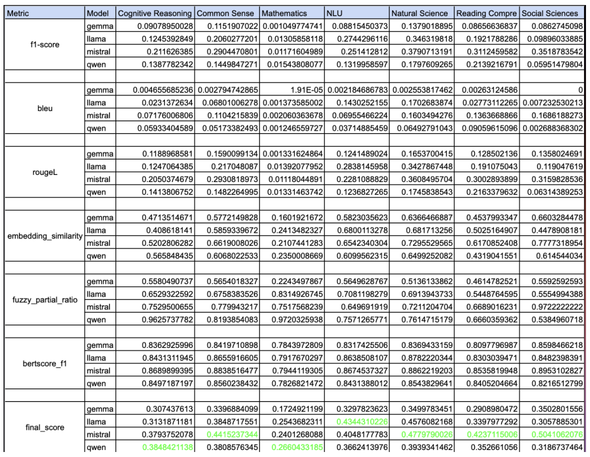


Figure 3: Architecture diagram of the Intelligent Planner



Metric	Model	Cognitive Reasoning	Common Sense	Mathematics	NLU	Natural Science	Reading Compre	Social Sciences
f1-score	gemma	0.09078950028	0.1151907022	0.001049774741	0.08815450373	0.1379018895	0.08656636837	0.0862745098
	llama	0.1245392849	0.2060277201	0.01305858118	0.2744296116	0.346319818	0.1921788286	0.09896033885
	mistral	0.211626385	0.2904470801	0.01171604989	0.251412812	0.3790713191	0.3112459582	0.3518783542
	qwen	0.1387782342	0.1449847271	0.01543808077	0.1319958597	0.1797609265	0.2139216791	0.05951479804
bleu	gemma	0.004655685236	0.002794742865	1.91E-05	0.002184686783	0.002553817462	0.00263124586	0
	llama	0.0231372634	0.06801006278	0.001373585002	0.1430252155	0.1702683874	0.02773112265	0.007232530213
	mistral	0.07176006806	0.1104215839	0.002060363678	0.06955466224	0.1603494276	0.1363668866	0.1686188273
	qwen	0.05933404589	0.05173382493	0.001246559727	0.03714885459	0.06492791043	0.09059615096	0.002688368302
rougeL	gemma	0.1188968581	0.1590099134	0.001331624864	0.1241489024	0.1653700415	0.128502136	0.1358024691
	llama	0.1247064385	0.217048087	0.01392077952	0.2838145958	0.3427867448	0.191075043	0.119047619
	mistral	0.2050374679	0.2930818973	0.01118044891	0.2281088829	0.3608495704	0.3002893899	0.3159828536
	qwen	0.1413806752	0.1482264995	0.01331463742	0.1236827265	0.1745838543	0.2163379632	0.06314389253
embedding_similarity	gemma	0.4713514671	0.5772149828	0.1601921672	0.5823035623	0.6366466887	0.4537993347	0.6603284478
	llama	0.408618141	0.5859339672	0.2413482327	0.6800113278	0.681713256	0.5025164907	0.4478908181
	mistral	0.5202806282	0.6619008026	0.2107441283	0.6542340304	0.7295529565	0.6170852408	0.7777318954
	qwen	0.565848435	0.6068022533	0.2350008669	0.6099562315	0.6499252082	0.4319041551	0.614544034
fuzzy_partial_ratio	gemma	0.5580490737	0.5654018327	0.2243497867	0.5649628767	0.5136133862	0.4614782521	0.5592592593
	llama	0.6529322592	0.6758383526	0.8314926745	0.7081198279	0.6913943733	0.5448764595	0.5554994388
	mistral	0.7529500655	0.779943217	0.7517568239	0.649691919	0.7211204704	0.6689016231	0.9722222222
	qwen	0.9625737782	0.8193854083	0.9720325938	0.7571265771	0.7614715179	0.6660359362	0.5384960718
bertscore_f1	gemma	0.8362925996	0.8419710898	0.7843972809	0.8317425506	0.8369433159	0.8097796987	0.8598466218
	llama	0.8431311945	0.8655916605	0.7917670297	0.8638508107	0.8782220344	0.8303039471	0.8482398391
	mistral	0.8689899395	0.8838516477	0.7944119305	0.8674537327	0.8862219203	0.8535819948	0.8953102827
	qwen	0.8497187197	0.8560238432	0.7826821472	0.8431388012	0.8543829641	0.8405204664	0.8216512799
final_score	gemma	0.307437613	0.3396884099	0.1724921199	0.3297823623	0.3499783451	0.2908980472	0.3502801556
	llama	0.3131871181	0.3848717551	0.2543682311	0.4344310226	0.4576082168	0.3397977292	0.3057885301
	mistral	0.3793752078	0.4415237344	0.2401268088	0.4048177783	0.4779790026	0.4237115006	0.5041062076
	qwen	0.3848421138	0.3808576345	0.2660433185	0.3662413976	0.3939341462	0.352661056	0.3186737464

Figure 4: Performance comparison of four LLMs (Gemma, Llama, Mistral, Qwen) across eight task domains

6 Experiments and Evaluation

We tested our routing mechanism using four popular benchmark datasets to assess model performance across diverse domains. SQuAD asked reading comprehension questions, MMLU tested general knowledge from science and technology, religion, and diet, PIQA asked common sense reasoning questions, and GSM8K tested mathematical problem-solving ability. This selection gave dense coverage of a range of cognitive tasks. Our evaluation required a combination of different evaluation techniques to cover various aspects of model performance. We counted on the usual metrics of F1-score for precision and recall evaluation, BLEU scores to measure n-gram overlap with reference texts, and RougeL scores to measure sequence similarity. To understand semantic quality, we added embedding similarity measurements to gauge how closely model responses matched ground truth in vector space. We also incorporated fuzzy partial ratio calculations to handle string variations and BERTScore F1 to analyze contextual meaning.

Each query went through systematic evaluation: models in our library processed identical inputs, then we collected their answers and reasoning. We scored these outputs using our metric suite and in comparison to ground truth data. After aggregating scores by task category, there were evident trends in performance (Figure 4). Mistral performed best on reading comprehension (0.4237) and natural sciences (0.4780), while Qwen performed best in mathematics (0.2660) and cognitive reasoning (0.3848). These scores directly affected routing configuration to guide queries to the top-performing model in a given category.

7 Results

The Django framework was used to build the user interface for the Intelligent Planner. The system features a chatbot-style interface, enabling users to interact naturally with the underlying language models. In addition to automatic model selection via the planner, user can manually select the model if he has any specific preference. This feature is accessible through a drop-down menu mentioning all the model's name and ofcoure the Planner as shown in Figure 5.

Depending on the type of query, we get the query type, model name, and then the query response as an output from the model, as shown in the Figure 6. This response's context is successfully saved and passed on when querying another model, no matter which model it chosen next. There is also a side panel that shows titles of previous chats. The

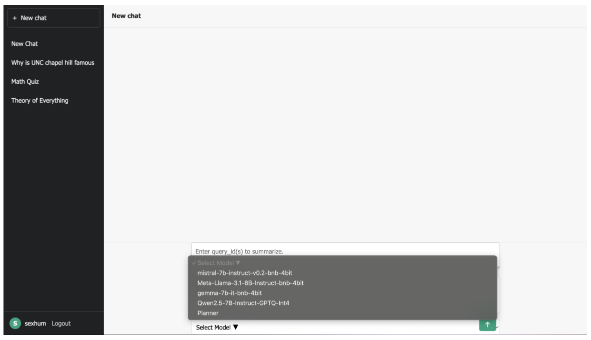
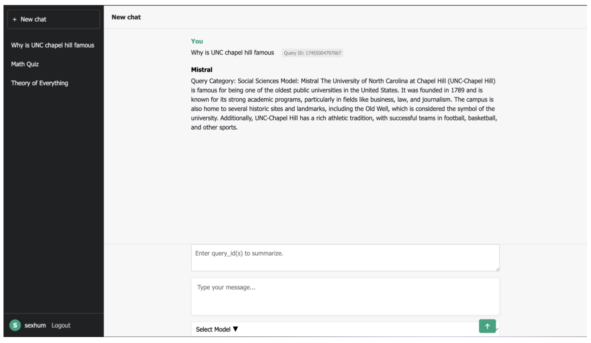

+ New chat
New chat
New Chat
Why is UNC chapel hill famous
Math Quiz
Theory of Everything
Enter query_id(s) to summarize.
✓ Select Model ▼
mistral-7b-instruct-v0.2-bnb-4bit
Meta-Llama-3.1-8B-Instruct-bnb-4bit
gemma-7b-it-bnb-4bit
Qwen2.5-7B-Instruct-GPTQ-Int4
Planner
Select Model ▼
S sexhum Logout

Figure 5: Model Selection Drop-down Menu



You

Why is UNC chapel hill famous Query ID: 17455004797067

Mistral

Query Category: Social Sciences Model: Mistral The University of North Carolina at Chapel Hill (UNC-Chapel Hill) is famous for being one of the oldest public universities in the United States. It was founded in 1789 and is known for its strong academic programs, particularly in fields like business, law, and journalism. The campus is also home to several historic sites and landmarks, including the Old Well, which is considered the symbol of the university. Additionally, UNC-Chapel Hill has a rich athletic tradition, with successful teams in football, basketball, and other sports.

Figure 6: Query Response from Planner with Query Category and Model used

text field provided to add a query ID can retrieve specific memory, which uses the query ID to even provide context across chats. This enables seamless transitions between conversations and ensures continuity in multi-turn interactions, regardless of the model used.

As shown in Figure 6, the user queries: *"Why is UNC Chapel Hill famous?"* The Intelligent Planner correctly classifies this question under the **Social Sciences** category. Based on prior experimental results (refer to Figure 4), the Mistral model demonstrated the highest accuracy for Social Sciences queries. Accordingly, the system routes the query to **Mistral**, which generates a coherent and informative response. The answer highlights UNC-Chapel Hill's historical significance, strong academic programs, and rich athletic tradition. This example validates the effectiveness of the classification and routing logic — not only was the domain identified accurately, but the best-suited model was also selected dynamically to deliver a high-quality answer.

8 Future Work

The current system works efficiently on 4-bit quantized models but there is scope of improvement given the resources. Few directions where this project can be extended include:

- Including VLLMs to include Image related queries.
  Many real-world applications of LLM require understanding of images, diagram or charts. Queries like visual question answering and diagram understanding would be routed to Visual-capable Large Language Models.

- Incorporating Advanced Models like ChatGPT, DeepSeek, Claude, etc
  The current architecture is constrained to use open-source, 4-bit quantized models, but for it to be really impactful, inclusion of state-of-the-art models like ChatGPT, DeepSeek, CLaude using their APIs would be crucial.

- Dataset Expansion and Task Diversity
  Currently there are 8 different broad classifications of query. To improve this classification granularity, variety of query types can be added for example code generation and instruction following. Expanding of data-set will aid in creating this and also help planner to be adequate in the reak world.

- Improving latency and make it more scalable
  To further improve latency, we can implement some run-time optimizations such as request batching and faster model loading/compilation.

References

[1] Pingzhi Li, Prateek Yadav, Jaehong Yoon, Jie Peng, Yi-Lin Sung, Mohit Bansal, and Tianlong Chen. 2024. Glider: Global and local instruction-driven expert router. *Preprint*, arXiv:2410.07172.

[2] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Preprint*, arXiv:2305.18290.

[3] Robert Schaeffer. 2024. How to fine-tune llama 3.1 (8b).

[4] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Preprint*, arXiv:2303.17580.

[5] Tong Xiao and Jingbo Zhu. 2025. Foundations of large language models. *Preprint*, arXiv:2501.09223.

[6] Prateek Yadav, Colin Raffel, Mohammed Muqeeth, Lucas Caccia, Haokun Liu, Tianlong Chen, Mohit Bansal, Leshem Choshen, and Alessandro Sordoni. 2024. A survey on model merging: Recycling and routing among specialized experts for collaborative learning. *arXiv preprint arXiv:2408.07057*.