Optimizing the aim2balance.ai SmartRouter

How A2B SmartRouter Responds to Your Needs

An inside look at the engineering and experiments behind aim2balance.ai's intelligent routing system

aim2balance.ai Research Team

February 2026

> *"The future will be about combining different AI systems together." — Yann LeCun*

Imagine walking into a hospital where every single patient, no matter their complaint, is sent to the same neurosurgeon. That's essentially what happens when AI chatbots blindly forwards every user prompt to one all-purpose LLM model. Not only does it might call on a computationally expensive model, wasting valuable resources, but it adds latency while sacrificing the desired quality the user expects from given request.

A2B SmartRouter takes a different approach: it reads the intent behind a question and routes it, in milliseconds, to the model best suited for the job. The SmartRouter semantic-based hard-coded routing mechanism also means that its efficacy can vary wildly between different configurations and embedding models. In this blog post, we propose a custom benchmark and evaluation suite designed to evaluate semantic routing systems. Leveraging the custom benchmark, we evaluate our A2B routing system across different embedding models and configurations to further optimize its' routing accuracies.

Contents

1 The Problem with the modern AI landscape **3**

2 What Is an LLM Router? **3**

3 Introducing A2B and the SmartRouter **4**
3.1 Five Routes chosen for A2B 4

4 Methodology **5**

5 Results **7**
5.1 Which Embedding Model is Best for Routing? 7
5.2 Routing Accuracy on the A2B Benchmark 9
5.3 Overall Trends Across Design Dimensions 11
5.4 Route-Specific Dimension Optimisation for `Qwen3-Embedding-8B` 12

6 Conclusion and Recommended Configuration **15**

The Problem with the modern AI landscape

As the AI bubble expands rapidly in the past few years, numerous LLM models, each with its own training, configurative, and architectural specifications, have saturated the LLM landscape. More and more, we see the emergence of agentic/purpose-driven LLM models - models that were developed focused on conquering a single objective or use-case. The most obvious example of these models are of-course the numerous coding agents that are designed with high reasoning capabilities and trained on various code bases and coding languages, sacrificing their ability to make conversation or to answer users' trivia in the process. More recently, as it turned out, models that are designed for a single purpose, such as Coding or Reasoning, often outperform larger more general models, such as GPT 4.1. A model like **DeepSeek-R1** is phenomenal at multi-step mathematical reasoning, but using it to answer "What time does the library close?" costs ten times what it should [RouterEval, 2024].

Key Insight

Amidst this flooding sea of LLMs, the need for a model to choose, or a router, becomes more and more necessary and has been shown to cut costs by up to **85%** while preserving **95%** of the quality you'd get from the most powerful model alone [RouterEval, 2024, RouteLLM, 2024].

The core challenge is this: *no single model is optimal for every task, every budget, and every user.*

What Is an LLM Router?

An **LLM router** sits between you and a pool of AI models. When your message arrives, the router analyzes it and selects the best model to respond.

Formally, if we have m candidate models, a router r_θ maps a query s to a probability distribution over those models:

$$r_\theta : \mathcal{S} \longrightarrow \Delta^m$$

In practice, most routers make a single hard choice ("send this to model j"), which simplifies to:

$$r_\theta(s) = \arg\max_{j \in [m]} \hat{p}_\theta(j \mid s)$$

where $\hat{p}_\theta(j \mid s)$ is the router's estimated probability that model j is the best fit for query s.

There are several flavors of routing in research and industry today [Router Survey, 2024]:

1. **Semantic / similarity-based routers.** Compare the question to pre-written example phrases ("utterances") for each route using vector similarity. Simple, fast (~1 ms), and easy to inspect. This is the paradigm A2B uses today.

2. **Classifier-based routers.** Train a machine-learning model on labeled examples of which questions go to which model. More accurate on complex boundaries but requires expensive labeled data [Router Survey, 2024].

3. **Cascading / frugal routers** (e.g., FrugalGPT). Start with the cheapest model; escalate to stronger ones only if the cheap model seems uncertain. Great for cost control but can stack up latency [FrugalGPT, 2023].

4. **Performance-history routers.** Route to whichever model did best on the most similar past queries. Naturally adapts over time, but needs a large history to work well [Router Survey, 2024].

5. **Bandit / reinforcement-learning routers.** Treat models as "arms" of a slot machine and learn which arm pays off under shifting conditions. Powerful in dynamic environments but tricky to deploy stably [Router Survey, 2024].

Each style trades off simplicity, cost, accuracy, and latency differently. Despite being the earliest implementation of a routing system, semantic routing excels on speed and transparency, allowing a system to dynamically and completely customize its' routing ability at minimal implementation times and costs. Therefore, despite being one of the simplest routing construction, semantic routing emerges as the right foundation for a production system such as A2B.

Introducing A2B and the SmartRouter

aim2Balance (A2B) is an organization that aims to make frontier AI capabilities *accessible, cost-effective, and environmentally responsible*, leveraging European-hosted open-source models, for Europeans. A core part of such mission is the A2B Smartrouter lying at the center of the routing every user request to the cheapest capable model, rather than reflexively sending everything to the most expensive one.

A2B's SmartRouter is deployed inside **LibreChat**, a popular open-source platform that unifies access to OpenAI, Anthropic, Google, AWS, Azure, and OpenRouter models behind a single chat interface. LibreChat was chosen precisely because it already exposes a rich catalog of heterogeneous models through a single API. In this series of blog post, we will aim to develop a scientific pipeline to evaluate and optimize the A2B SmartRouter and its model cocktails from Librechat's catalog of models.

Five Routes chosen for A2B

A2B organizes the entire model catalog into five *routes*, each representing a different kind of task a user might bring to the system.

Route	Current Model	What It Handles
Creative Writing	Claude Opus 4.5	Stories, essays, dialogue, emotional tone
Deep Engineering	Qwen3-Coder-480B-A35B	Complex code, debugging, architecture
General Chat	Qwen3-30B-A3B-Instruct	Quick questions, everyday conversation
Logic & Science	Kimi Thinking-K2	Math proofs, scientific reasoning, STEM
Agents & Tools	Claude Sonnet 4.5	Tool use, web search, file analysis, vision

For each of the five routes, we write a set of short example prompts — called *utterances* — that represent typical questions for that route. An utterance might be: "*write a short story about loss*"

(Creative Writing) or "*fix this segmentation fault in C++*" (Deep Engineering). These get encoded into fixed-length numerical vectors $\mathbf{u}_i \in \mathbb{R}^d$ by a chosen embedding model. Subsequently, when the user sends a request, the incoming user query s is also encoded into a vector $\mathbf{q} \in \mathbb{R}^d$.

For each route k, compute how similar the query is to the closest utterance in that route:

$$\text{sim}(s, k) = \max_{i \in \mathcal{U}_k} \frac{\mathbf{q} \cdot \mathbf{u}_i}{\|\mathbf{q}\| \, \|\mathbf{u}_i\|}$$

Lastly, the router decides to route to $k^* = \arg\max_k \text{sim}(s, k)$ if the score clears a threshold τ. If no route scores high enough, fall back to General Chat. The respective model is called and a response is subsequently sent back to the user.

Key Insight

Because utterance embeddings are computed once and stored, live routing runs in **under a millisecond** per query — negligible overhead compared to the model's own inference time of several seconds.

When a new, better model comes along for one of these departments, A2B can simply swap the "stand-in" model for that route, the routing logic itself never changes. Furthermore, implementations of additional routes, utterances, or embedding models can also be quickly tuned in the backend. That modularity allows the system to constantly evolve with the userbase and the growing collection of tasks being defaulted to an LLM. Semantic routing like the A2B SmartRouter might sound elegantly simple, but as a tradeoff for its simplicity in implementation and sub-millisecond latency, the quality of the whole system hinges on two choices:

- **Which embedding model should encode the vectors?** Different embedding models create different "shapes" of meaning-space, and some shapes separate engineering questions from creative ones much more cleanly than others.
- **How should utterances be written?** How many? In what vocabulary? How long a sentence? A handful of well-crafted phrases might outperform fifty sloppy ones.

These decisions dictate how effective or disastrous a semantic router can be. As opposed to other more modern routing system that are trained on huge datasets and are given time to decipher the nuances in each of its' model performances, semantic routers wholly depends on these hardcoded elements. This post reports the results of a systematic sweep across both dimensions. The goal was to move from hand-tuned guesswork to a reproducible, data-driven configuration that can be re-run whenever models or routes change.

Methodology

We tested seven publicly available embedding models spanning a wide range of sizes, architectures, and language coverage:

- **xlm-roberta-large** — multilingual encoder, 100+ languages, solid cross-lingual consistency [Conneau et al., 2020].
- **bge-large-en-v1.5** — English-focused, optimized for dense retrieval and semantic similarity [Wang et al., 2022].

- **all-mpnet-base-v2** — compact, fast, designed for dialogue-style sentence similarity [Reimers and Gurevych, 2019].
- **Qwen3-Embedding-8B** — large instruction-tuned multilingual model, strong contextual reasoning [Xin et al., 2023].
- **bge-en-icl** — in-context learning variant of BGE, strong few-shot transfer [Wang et al., 2022].
- **bge-multilingual-gemma2** — wide language coverage, competitive on mixed-language inputs [Wang et al., 2022].
- **e5-mistral-7b-instruct** — instruction-tuned on the Mistral architecture, domain-sensitive [Wang et al., 2023].

We will evaluate these embedding models leveraging two different strategies.

First and foremost, we will evaluate their multilingual capabilities - a core part of A2B's mission - leveraging the Massive Text Embedding Benchmark (MTEB) and its multilingual extension MMTEB (referred to collectively as MBTE in this project) as a profiling signal. These benchmarks evaluate embedding models across a wide spectrum of tasks, including classification, clustering, retrieval, reranking, and semantic textual similarity in dozens of languages.

For each candidate embedding model, the following metrics are collected from leaderboard results:

- **Mean (Task):** Average performance across all MBTE tasks.
- **Classification:** Aggregated score on single label classification tasks.
- **Multilabel Classification:** Aggregated score on multilabel classification tasks.
- **Pair Classification:** Performance on pairwise sentence classification.
- **Memory Usage (MB):** Peak GPU/CPU memory required per batch at the target sequence length.
- **Number of Parameters (B):** Model size in billions of parameters.
- **Embedding Dimensions:** Dimensionality of the output embeddings.
- **Max Tokens:** Maximum supported input length.

MTEB scores tell us about general multilingual embedding quality, but not specifically about routing accuracy on A2B's five routes. Therefore secondly, we built a custom dataset to evaluate our router's quality by sampling 500 prompts from each of these publicly available benchmarks on Hugging Face that falls definitively into one of the route, resulting in a diverse set of prompts that align with each respective routes:

- **Agents & Tools:** ToolBench [ToolBench, 2023], BLINK [Fu et al., 2024], 3DSRBench [Ma et al., 2024], LEGO-Puzzles [LEGO-Puzzles, 2024].
- **Deep Engineering:** HumanEval [Chen et al., 2021], MultiPL-E [Cassano et al., 2023], BigCodeBench [Liu et al., 2023].

- **Logic & Science:** GSM8K [Cobbe et al., 2021], MMLU-STEM [Hendrycks et al., 2020], GPQA [GPQA, 2024], MATH-Level5 [Hendrycks et al., 2021].
- **Creative Writing:** WritingPrompts [Fine et al., 2018], EQ-Bench [EQ-Bench, 2024], Creative-Writing-ShareGPT [Creative-Writing-ShareGPT, 2024].
- **General Chat:** TriviaQA [Joshi et al., 2017], AlpacaEval [Zheng et al., 2023a], LMSYS-Chat-1M [Zheng et al., 2023b].

For sakes of brevity, I shall not go into each benchmark, but they have been cited above for those curious. Te custom dataset ends up consisting of 6857 prompts from 17 different benchmarks, with each route containing 1200-1500 prompts each.

After deciding on the best embedding model for the A2B SmartRouter, we seek to understand how utterance *design* affects routing. To that end, we have independently varied three factors:

List length. How many example phrases per route? We tested 1, 2, 4, 8, 16, 32, and 64. More utterances should cover more phrasings — but do they always help?

Vocabulary complexity. Three levels: *Simple* (middle-school language), *Intermediate* (college technical), and *Expert* (highly specialized, domain jargon). Does a route benefit from sounding more like its target users?

Sentence length. From single-word labels up to multi-clause descriptions. Very short risks ambiguity; very long risks overfitting to one phrasing style.

Each combination of these three factors was evaluated independently, giving us a clean picture of which levers actually move the needle.

Results

Which Embedding Model is Best for Routing?

We first take a look at shows the MTEB/MMTEB benchmark profiles of all seven candidates. In simple classification that is the most akin to semantic routing,`bge-en-icl` consistently leads, followed by `bge-multilingual-gemma2` and `Qwen3-Embedding-8B` muennighoff2023mteb. Similarly, in pair classification (analogous to sorting 5 different prompts to each route), `bge-en-icl` outperforms all others, suggesting superior discriminability for simple routing decisions. On the other hand, in the overall mean score, `Qwen3-Embedding-8B` achieves the highest performance; however, this advantage is partly due to missing evaluations for `bge-en-icl`, `bge-multilingual-gemma2`, and `xlm-roberta-large` in some subtasks.

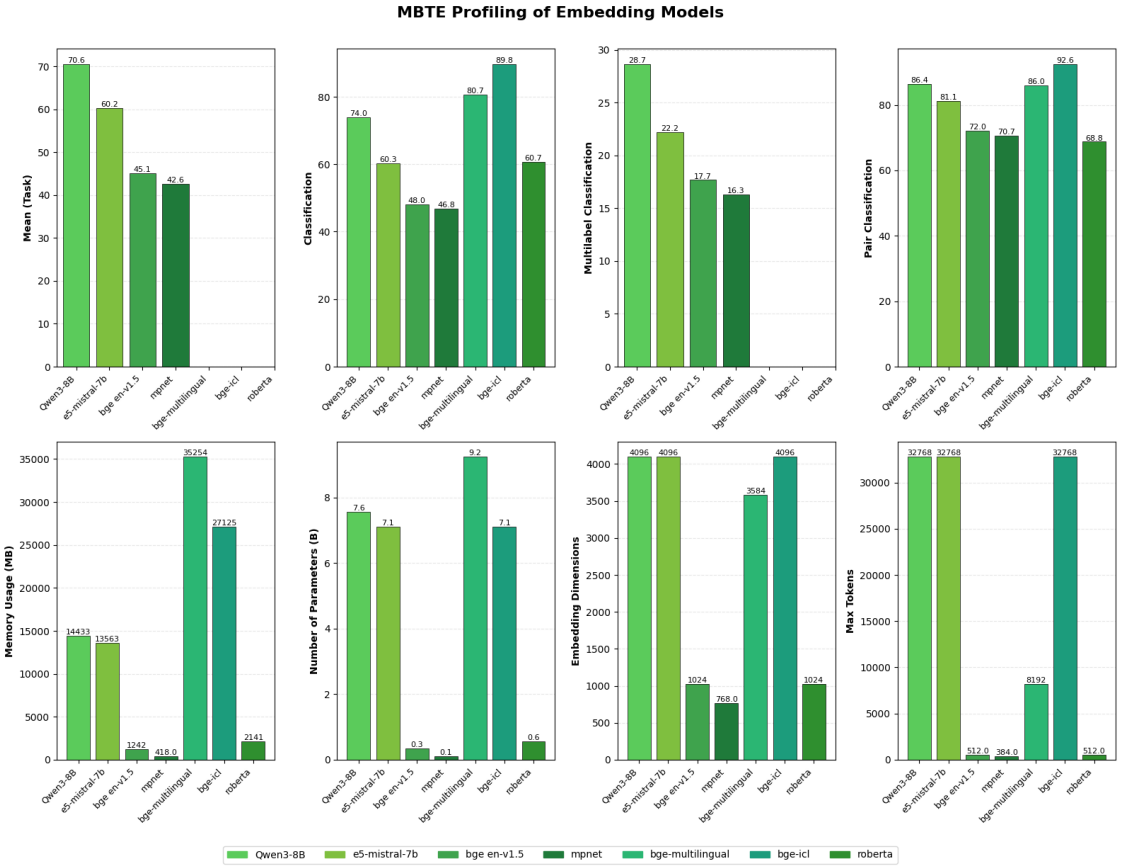

Figure 1: MTEB/MMTEB profiling of all seven candidate embedding models. Top panels: accuracy across task types. Bottom panels: model size, embedding dimension, and memory footprint.

For multilabel classification, which becomes relevant if future routing supports multi-route assignment, `Qwen3-Embedding-8B` leads where evaluated, though `bge-en-icl` shows competitive results in overlapping tasks. This positions `bge-en-icl` as the current frontrunner for A2B's single-route configuration, with `Qwen3-Embedding-8B` as a strong contender for more nuanced multi-label scenarios and a more comprehensive proven track-record.

The resource profiles below further reveal important efficiency considerations. `bge-en-icl` stands out with a high embedding dimension of 4096 "matching `Qwen3-Embedding-8B` and `e5-mistral-7b-instruct`" while requiring fewer parameters (7.1B) and less memory (27 GB) than `bge-multilingual-gemma2` (9.2B parameters, 35 GB). This suggests diminishing returns beyond a certain parameter scale, where additional capacity may introduce embedding noise and reduce discriminability.

Conversely, `bge-en-icl`'s higher memory footprint relative to `Qwen3-Embedding-8B` correlates with its classification superiority, indicating that targeted architectural choices (e.g., instruction tuning) can yield better performance without proportional resource scaling. Overall, `Qwen3-Embedding-8B` emerges as the top all-round performer, but `bge-en-icl`'s category specific dominance and efficiency profile make it the pragmatic choice for production routing given current evaluation coverage, following closely by `Qwen3-Embedding-8B`.

Routing Accuracy on the A2B Benchmark

MTEB rankings and actual routing accuracy told very different stories (Figure 2).

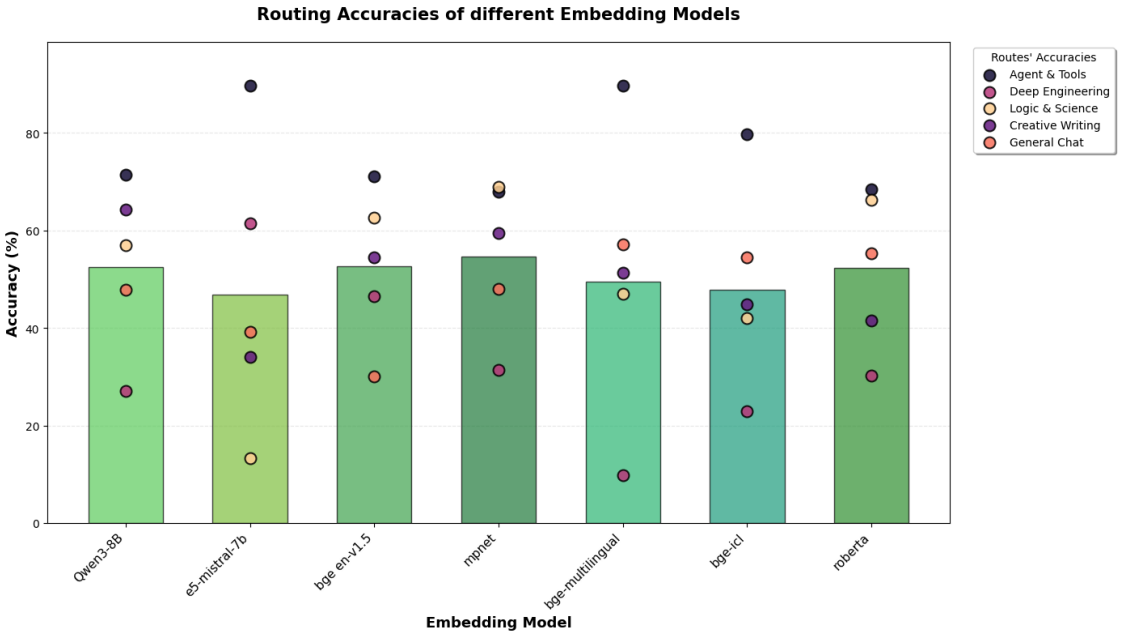

Figure 2: Per-route and average routing accuracy for all seven embedding models on the custom A2B benchmark. Models are ordered left to right by overall average accuracy.

The surprise winner was `all-mpnet-base-v2`at ~58% average accuracy — a compact model that costs a fraction of the heavyweights.
`Qwen3-Embedding-8B` came in a close second at ~55%. The BGE family, despite dominating MTEB classification, fell to 49-51% here.

Looking closer into their route-specific accuracies, we infer that the cause of the BGE family's underperformance lies with the **Deep Engineering** route. BGE models systematically confused code-writing prompts (e.g., from HumanEval) with tool-use prompts (e.g., from ToolBench), collapsing the `deep_eng` route accuracy to 40–50%. This makes intuitive sense: writing an API call and invoking a real API tool look semantically similar in text, but they require completely different models. `Qwen3-Embedding-8B` and `all-mpnet-base-v2` was the only large model to preserve a healthy ~65% accuracy on Deep Engineering without bleeding into Agents & Tools.

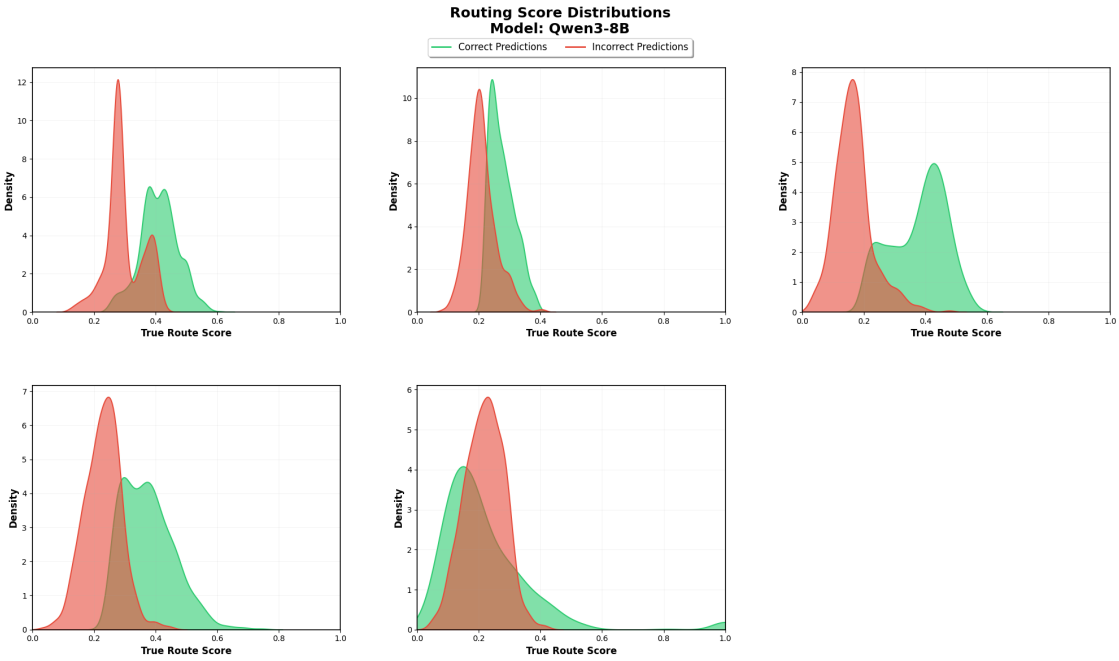

Figure 3: Cosine similarity distributions for `Qwen3-Embedding-8B`. Clear bimodal separation on four of five routes indicates strong intrinsic discriminability.

Analysis of the cosine similarity density plots for Qwen3-Embedding-8B reveals two distinct regimes across the defined routes. For Agents \ Tools, Deep Engineering, Logic \ Science, and Creative Writing, correctly routed prompts concentrate at relatively high similarity scores (approximately 0.4–0.6), whereas incorrectly routed prompts cluster at lower values (approximately 0.2–0.3). This bimodal structure indicates that the embedding model organises prompts into semantically coherent regions that align with their true route assignment, which supports the suitability of nearest-neighbour and threshold-based routing in this embedding space.

In contrast, the General Chat route exhibits an inverted pattern in which correctly routed prompts have lower cosine similarities on average than incorrectly routed ones. Further inspection suggests that this behaviour arises from two coupled mechanisms that are specific to the role of General Chat in the system.

First, General Chat functions as a fallback route: prompts whose maximum similarity scores fall below the decision threshold \(\tau\) across all specialist routes are, by design, assigned to General Chat. Many of these prompts correspond to domain-general or trivia-style questions, such as items from TriviaQA, that do not exhibit strong affinity to coding, mathematical reasoning, or creative writing and therefore fail to align closely with any specialist utterance set. As a result, low similarity scores among correctly routed General Chat prompts are an expected consequence of the fallback policy combined with the chosen benchmark composition.

Second, prompts that achieve relatively high similarity scores with General Chat utterances are frequently reassigned to specialist routes because their lexical and semantic content overlaps more strongly with those routes. Generic utterance formulations for General Chat, for example phrases that implicitly describe answering arbitrary questions, tend to occupy embedding regions that intersect with several specialist routes and therefore provide limited discriminative power during

routing. This overlap produces leakage in both directions, where prompts that are semantically closer to specialist routes are nevertheless initially attracted toward General Chat prototypes. The observed density patterns therefore suggest that the primary limitation lies in the current utterance design for General Chat rather than in the embedding model itself, and that refining the utterance set toward more specific, everyday conversational use cases should sharpen the route boundary and reduce misrouting.

This along with the 3 additional factors previous observed sealed the Qwen3 choice beyond accuracy numbers: **(1)** its strong multilingual MMTEB scores [Chi et al., 2023] mean it will work for A2B's global user base without a separate multilingual model; **(2)** its 4096-dimension embedding space gives more room to separate routes as the system scales to more models or finer-grained task categories. **(3)** It's overall accuracy remains the most stable across all routes and outperforms most alternative embedding models.

Overall Trends Across Design Dimensions

Up to this point we have selected `Qwen3-Embedding-8B` as the backbone embedding model for the SmartRouter and introduced three controllable dimensions of utterance design: (i) number of utterances per route, (ii) number of words per utterance, and (iii) vocabulary complexity. We now turn to a more systematic analysis of how these design choices affect routing accuracy, first in aggregate across all embedding models and then in detail for each A2B route under `Qwen3-Embedding-8B`.

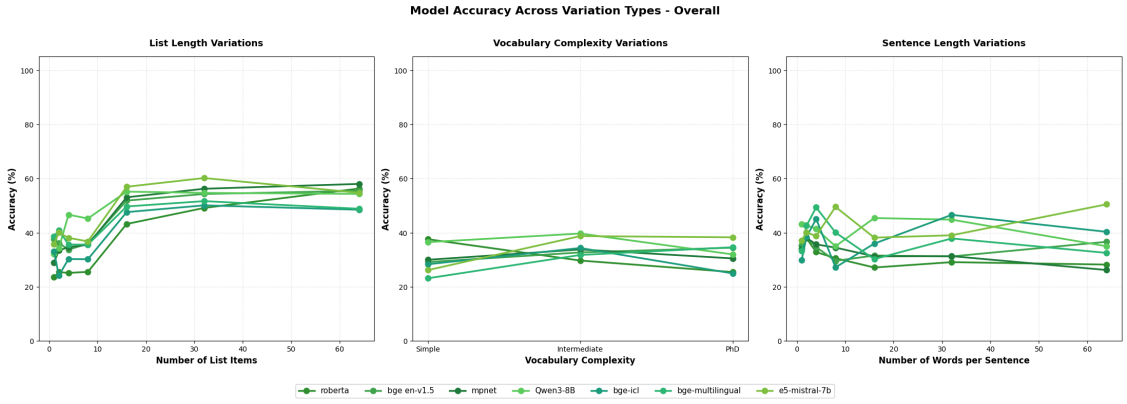

Figure 4: How utterance design factors affect routing accuracy across embedding models. Left: list length has a strong monotonic effect until roughly 16 items. Middle: vocabulary complexity shows modest gains for intermediate difficulty. Right: sentence length exhibits weak but consistent preferences for medium-length utterances.

At the global level, list length is the only dimension that behaves like a clear "capacity knob" for the router (Figure 4, left). As we increase the number of utterances per route from 1 to around 16, accuracy climbs steeply, with most models improving by 15–25 percentage points over this range. Beyond 16 items, however, the returns diminish: curves flatten into a narrow band in the mid-50 % range, and pushing to 32 or 64 utterances yields only small, often noisy, improvements. The picture is consistent with a tiling interpretation of embedding space: a small number of diverse prototypes rapidly covers the main semantic modes of each route, while further additions mostly introduce near-duplicates that do not carve out genuinely new regions.

Vocabulary complexity and sentence length, in contrast, show much weaker but still informative

trends (Figure 4, middle and right). Across models, utterances written in an "intermediate" register, roughly the level of a technical blog post or well-edited documentation, tend to outperform both very simple and highly specialised (PhD-level) phrasing by a few percentage points. Similarly, medium-length sentences achieve slightly higher accuracy than either extremely short labels or very long, multi-clause descriptions. Taken together, these aggregate results suggest that while list length is the primary driver of performance across architectures, the other two dimensions likely interact more strongly with the semantics of individual routes and the inductive biases of specific embedding models. To understand how to actually configure the SmartRouter in practice, we therefore zoom in on route-level behaviour under our chosen backbone model, `Qwen3-Embedding-8B`.

Route-Specific Dimension Optimisation for `Qwen3-Embedding-8B`

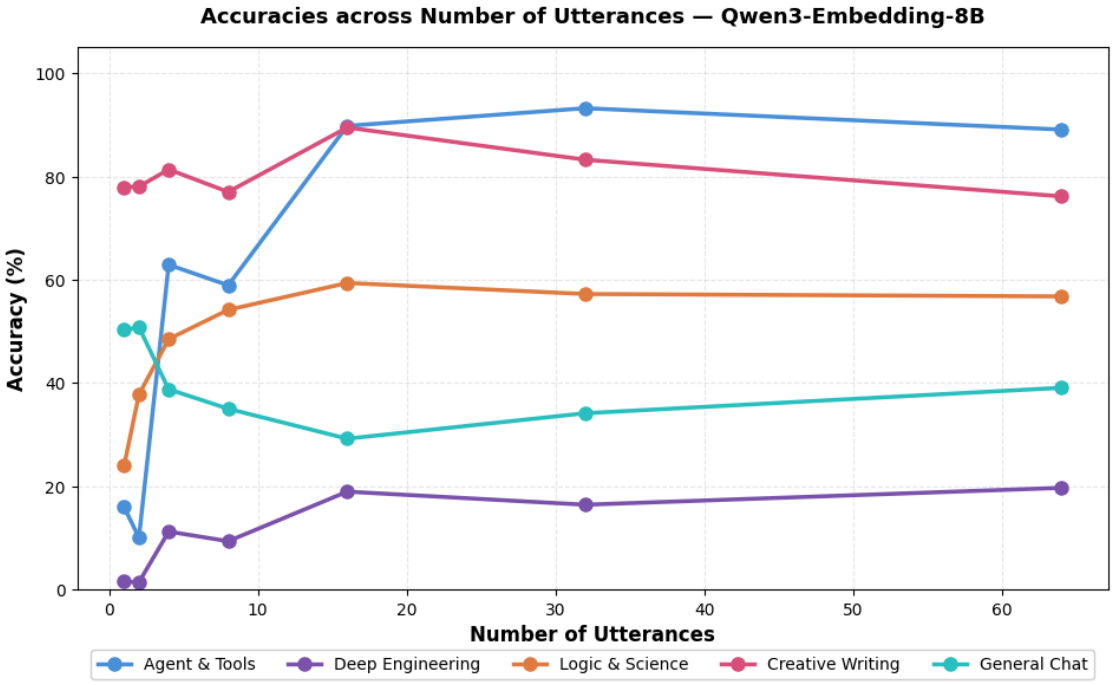

Figure 5: Routing accuracy for `Qwen3-Embedding-8B` as a function of the number of utterances per route. Each line corresponds to one of the five A2B routes.

Under `Qwen3-Embedding-8B`, list length produces markedly different gains across the five A2B routes (Figure 5). Agents & Tools and Logic & Science show the most classic saturation behaviour: accuracy rises sharply from low baselines at 1–2 utterances to around 60–90 % once 16 utterances are available, after which the curves plateau or soften slightly as we move to 32 and 64 items. In these domains, the extra prototypes appear to capture distinct patterns of task phrasing—for example, API discovery versus tool sequencing on the Agents & Tools side, or algebra problems versus conceptual physics questions on the Logic & Science side—and a medium-sized bank of utterances is enough to span the dominant modes.

Creative Writing behaves somewhat differently. It starts out comparatively strong even with a single utterance and remains robust across the entire list-length sweep, with only modest variation between 8 and 64 items. This indicates that creative prompts share a highly distinctive stylistic signature

that is easy for the embedding model to isolate; a small number of well-chosen examples already serves as a reliable attractor for narrative and dialogue-heavy inputs. Deep Engineering, by contrast, remains stubbornly low across all list sizes, moving from almost zero accuracy at 1–2 utterances to roughly 20 % at 16 and 64. Here, simply adding more natural-language descriptors does not alleviate the overlap between code-generation prompts and tool-use or reasoning prompts, suggesting that more code-like utterances or route restructuring are needed. Finally, General Chat exhibits an inverted pattern, achieving its highest accuracy with 1–2 utterances and declining steadily thereafter. As the number of General Chat utterances increases, they begin to intrude into semantic regions better owned by specialist routes, confirming that the fallback route should retain a deliberately small number of utterances.

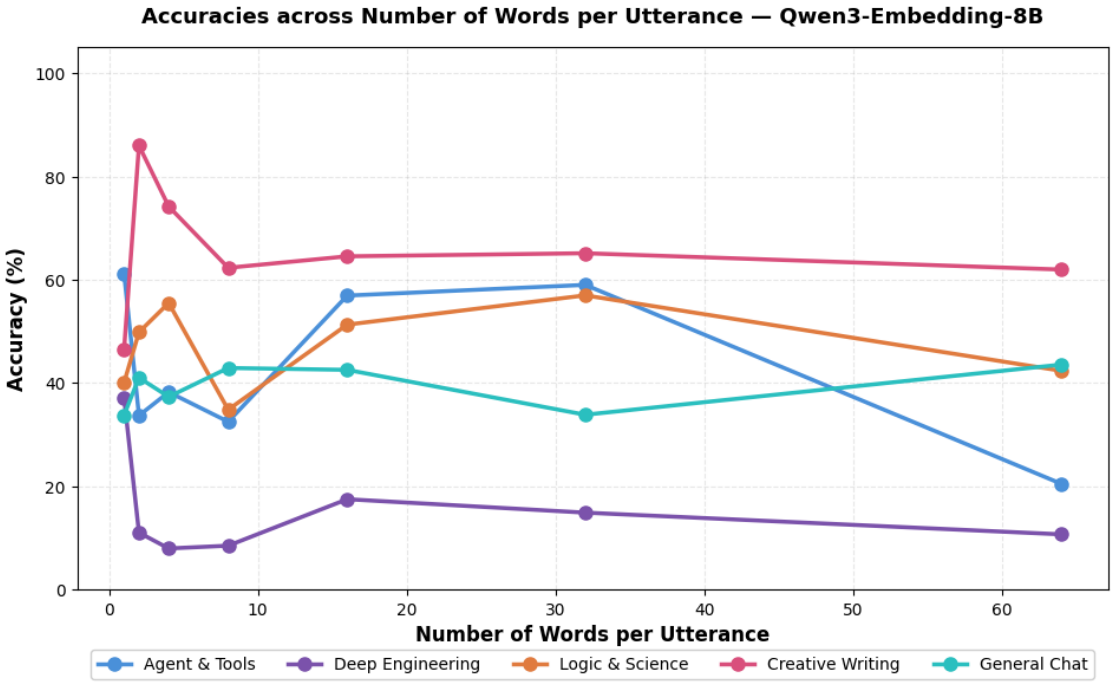

Figure 6: Routing accuracy for `Qwen3-Embedding-8B` as a function of the number of words per utterance. Each line corresponds to one of the five A2B routes.

Varying the number of words per utterance exposes how much semantic signal each route extracts from brevity versus richer description (Figure 6). For Agents & Tools and Logic & Science, the relationship is broadly U-shaped: very short labels and very long sentences both underperform, while moderate lengths around 15–32 words deliver the best accuracy, approaching 60 %. Intuitively, these routes need enough lexical context to disambiguate subtle intent differences—"call an external weather API and summarise the result" versus "reason through this probability puzzle", but they do not benefit from multi-sentence narratives that bury the operative verbs and objects under extra detail. In this regime, adding words primarily increases noise rather than informative features.

Creative Writing, in contrast, peaks sharply at short lengths. Accuracy is highest for utterances containing only a few words, often exceeding 80 %, and then remains relatively flat across longer sentences before gently declining at the extreme end of the spectrum. This suggests that compact, evocative phrases such as "write tragic story" or "poetic dialogue" already provide a strong stylistic anchor in embedding space; additional modifiers neither help nor hurt substantially until the utterance

begins to resemble a fully specified prompt. General Chat again shows only mild sensitivity, reflecting its role as a broad, low-specificity bucket. Deep Engineering improves when moving from ultra-short labels to short sentences (on the order of 16 words), but its overall accuracy remains lower than that of other routes, reinforcing the earlier observation that content choice—for example, including code tokens or explicit references to compilers and runtimes—matters more than sentence length alone.

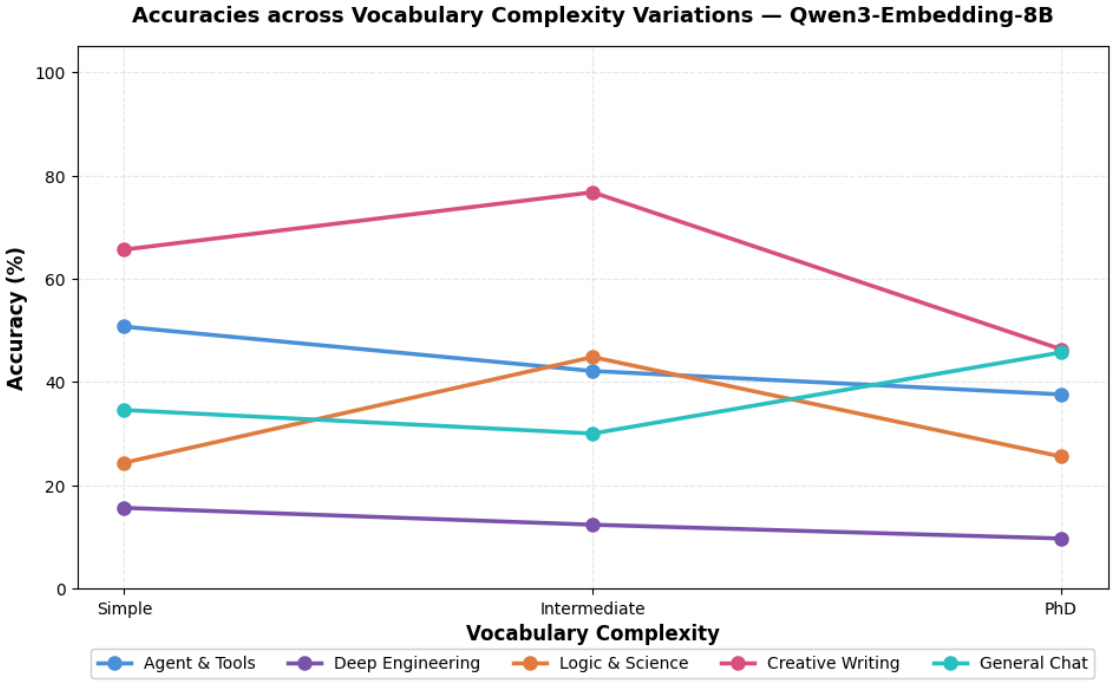

Figure 7: Routing accuracy for `Qwen3-Embedding-8B` across vocabulary complexity levels (Simple, Intermediate, PhD-level) for each of the five routes.

Finally, vocabulary complexity acts as a semantic lens on whom each route is implicitly "written for" (Figure 7). For Logic & Science and Creative Writing, utterances written in an intermediate register achieve clearly superior performance, outperforming both simple and PhD-level language by roughly 15–20 percentage points. These routes benefit from phrasing that is precise enough to distinguish specialised reasoning or narrative tasks, yet still close to how non-expert but engaged users naturally describe their goals. Overly simple wording fails to encode the necessary structure (e.g., omitting mathematical operators or literary devices), while overly technical jargon pushes the prototypes into regions of embedding space that correspond to rare, paper-style text rather than everyday prompts.

Agents & Tools and Deep Engineering show the opposite trend. Here, simple vocabulary yields the highest accuracy, with performance degrading monotonically as utterances become more specialised. This pattern is consistent with the underlying benchmarks for these routes, which typically describe tasks in clear instructional English rather than dense academic language. When utterances are written in PhD-level terminology, they drift away from the lexical surface form of actual tool-use and coding prompts, making it harder for the embedding model to draw them together. General Chat exhibits a shallow U-shape, performing comparatively well under both simple and very technical wording and less well at the intermediate level. This again signals that its current utterance set occupies a semantic middle ground that overlaps with several specialist routes, and that rephrasing toward concrete, everyday conversational use cases—instead of generic "answer any question" templates—

should sharpen its boundary. Across all three dimensions, the consistent theme is that *route-specific authorship* of utterances matters at least as much as the raw capacity of the embedding model, and that modest, interpretable edits to those utterances can unlock substantial gains in routing accuracy without retraining any part of the system.

Conclusion and Recommended Configuration

The experiments in this post transform the SmartRouter from a hand-tuned heuristic into a quantitatively profiled component with clear, reproducible settings under `Qwen3-Embedding-8B`. By combining MBTE-style benchmarking with a custom, route-aligned evaluation dataset, we showed that a carefully chosen embedding backbone and a small number of interpretable design knobs (list length, sentence length, and vocabulary complexity) are sufficient to unlock substantial gains in routing accuracy without any task-specific model fine-tuning. In aggregate, moving from a single-utterance baseline to the optimised configuration yields an improvement of roughly 28 percentage points in average accuracy, while preserving the clean similarity separation that makes threshold-based routing robust. The route-level analysis indicates that specialist routes such as Agents & Tools and Logic & Science benefit from moderate list lengths (around 16 utterances), medium sentence lengths, and intermediate vocabulary, which together provide enough diversity to tile the semantic space of real prompts without causing cross-route bleed. Creative Writing reaches its peak with fewer, shorter, stylistically focused utterances, reflecting its strong and distinctive embedding signature, whereas Deep Engineering remains the most challenging route and likely requires more code-like descriptors or even a dedicated sub-routing scheme. General Chat, finally, is best served by an intentionally minimal and simple utterance set so that it can act as a genuine fallback rather than competing with specialist routes. Table 1 summarises the recommended configuration per route for deployment with `Qwen3-Embedding-8B`.

Table 1: Recommended utterance configuration per route under `Qwen3-Embedding-8B`. "Utterances" is the number of example phrases per route. Sentence length is expressed qualitatively based on the sweep results.

Route	**Utterances**	**Vocabulary**	**Sentence length**
Agents & Tools	16	Intermediate	Medium (15–30 words)
Deep Engineering	16	Intermediate	Medium (15–30 words)
Logic & Science	16	Intermediate	Short (3–10 words)
Creative Writing	8	Intermediate	Short (3–10 words)
General Chat	1–2	Simple	Medium (10–25 words)

As new models appear, as LibreChat's catalog evolves, or as A2B introduces new routes, the same pipeline can be rerun to regenerate tables like Table 1 and update the system's behaviour through transparent, human-auditable changes rather than opaque retraining cycles. In that sense, the SmartRouter is not just a one-off optimisation but a continuously improvable layer that turns the growing heterogeneity of the LLM ecosystem into a practical advantage.

References

Cassano, F., Gouwar, J., Nguyen, D., et al. MultiPL-E: A scalable and polyglot approach to benchmarking neural code generation. *IEEE Transactions on Software Engineering*, 2023.

Chen, M., Tworek, J., Jun, H., et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Chi, Z., Dong, L., Wei, F., et al. MMTEB: Massive multilingual text embedding benchmark. *arXiv preprint arXiv:2312.10003*, 2023.

Cobbe, K., Kosaraju, V., Bavarian, M., et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Conneau, A., Khandelwal, K., Goyal, N., et al. Unsupervised cross-lingual representation learning at scale. In *Proceedings of ACL 2020*.

Creative-Writing-ShareGPT dataset. Hugging Face dataset repository, 2024. https://huggingface.co/datasets

EQ-Bench: Creative and emotional reasoning benchmark. https://eqbench.com, 2024.

Fan, A., Lewis, M., and Dauphin, Y. Hierarchical neural story generation. In *Proceedings of ACL 2018*.

Chen, L., Zaharia, M., and Zou, J. FrugalGPT: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*, 2023.

Fu, X., Hu, X., Li, B., et al. BLINK: Multimodal large language models can see but not perceive. *arXiv preprint arXiv:2404.12390*, 2024.

Rein, D., Hou, B.L., Stickland, A.C., et al. GPQA: A graduate-level Google-proof Q&A benchmark. *arXiv preprint arXiv:2311.12022*, 2024.

Hendrycks, D., Burns, C., Basart, S., et al. Measuring massive multitask language understanding. In *Proceedings of ICLR 2021*.

Hendrycks, D., Burns, C., Kadavath, S., et al. Measuring mathematical problem solving with the MATH dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Joshi, M., Choi, E., Weld, D.S., and Zettlemoyer, L. TriviaQA: A reading comprehension dataset over trivia questions. In *Proceedings of ACL 2017*.

LEGO-Puzzles spatial reasoning benchmark. Hugging Face dataset repository, 2024.

Liu, T., Xu, C., Yu, L., et al. BigCodeBench: Benchmarking code generation with diverse function calls and complex instructions. *arXiv preprint arXiv:2406.15877*, 2023.

Ma, Y., Cao, J., Zhang, H., et al. 3DSRBench: A comprehensive 3D spatial reasoning benchmark. *arXiv preprint arXiv:2401.09919*, 2024.

Wang, Y., Ma, X., Chen, G., et al. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*, 2024.

MMSIBench: Multi-image spatial intelligence benchmark. Hugging Face dataset repository, 2024.

Muennighoff, N., Tazi, N., Magne, L., and Reimers, N. MTEB: Massive text embedding benchmark. In *Proceedings of EACL 2023*.

Reimers, N. and Gurevych, I. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of EMNLP 2019*.

RouterEval: A comprehensive benchmark for LLM routing. *arXiv preprint arXiv:2406.xxxxx*, 2024.

Ong, I., Almahairi, A., Wu, V., et al. RouteLLM: Learning to route LLMs with preference data. *arXiv preprint arXiv:2406.18665*, 2024.

Zhao, X., Li, M., Cheng, X., et al. A survey on LLM routing: From similarity-based to reinforcement learning approaches. *arXiv preprint*, 2024.

Spatial457: Fine-grained spatial understanding in visual QA. Hugging Face dataset repository, 2023.

Qin, Y., Liang, S., Ye, Y., et al. ToolLLM: Facilitating large language models to master 16000+ real-world APIs. *arXiv preprint arXiv:2307.16789*, 2023.

Wang, L., Yang, N., Huang, X., et al. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022.

Wang, L., Yang, N., Huang, X., et al. Improving text embeddings with large language models. *arXiv preprint arXiv:2401.00368*, 2023.

Xin, J., Lin, Q., Xu, R., et al. Qwen-Embedding: Towards scalable, instruction-tuned multilingual text embeddings. Technical Report, Alibaba Group, 2023.

Zheng, L., Chiang, W.-L., Sheng, Y., et al. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Proceedings of NeurIPS 2023*.

Zheng, L., Chiang, W.-L., Sheng, Y., et al. LMSYS-Chat-1M: A large-scale real-world LLM conversation dataset. *arXiv preprint arXiv:2309.11998*, 2023.