BetterQuery Intention Bridge
Technical Summary

CDLI Lab
research@cdli.ai

Abstract

BetterQuery improves prompt enhancement by routing each request through an explicit *Intention Bridge* before generation. This reduces ambiguity, enables safer fallbacks, and makes system behavior observable through production telemetry.

1 What is the Intention Bridge?

Instead of sending every input directly to an LLM, BetterQuery first:

1. detects intent using rules and embeddings,
2. optionally resolves ambiguity using a routing model,
3. chooses a specialized skill or falls back safely.

2 Core Pipeline

request → sanitize → session context → route → cache → generate (architect + refiner) → quality gate → response + analytics.

3 Why it works

- **Speed:** two-level cache (exact hash + semantic match).
- **Quality:** two-stage generation and gating.
- **Reliability:** controlled fallback/degraded modes.
- **Measurability:** admin metrics for skill usage, latency, fallback, and activity.

4 What we can prove today

The system already exposes measurable signals:

- top skills by usage,
- fallback and degraded request ratio,
- average latency,
- daily active enhancers,
- explicit quality feedback (rating + thumbs signals).

5 Benchmark plan

We compare pipeline variants on a fixed benchmark:

- full system,
- no LLM arbitration,
- no cache,
- one-stage generation.

Primary outcomes: intent precision/recall, p50/p95 latency, cost proxy, fallback rate, and human preference agreement.

6 Conclusion

BetterQuerys architecture is strongest as a layered control system: route by intent, generate with skill constraints, gate quality, and learn from telemetry. For further improvement, the highest leverage lies in better routing calibration and richer benchmark discipline.