From Demo to Production: Generative Search Lessons from Alexa+

Jitesh Mehta
Amazon
Sunnyvale, CA, USA
mjitesh@amazon.com

Rahul Joshi
Amazon
Sunnyvale, CA, USA
rahjoshi@amazon.com

Adarshan Sivashunmugam
Amazon
Sunnyvale, CA, USA
sivashun@amazon.com

ABSTRACT

Generative, LLM-powered search demos beautifully, then collides with production: hard latency deadlines, a bill that scales with traffic, and near-zero tolerance for failure. This experience report argues that the hidden specification for production AI search is *the customer's expectation of difficulty*—how hard a request looks to them—which sets both the quality they demand and the latency they tolerate. Drawing on generative video search built for the launch of Alexa+, Amazon's next-generation assistant, across Fire TV and Echo Show serving millions of customers, we report six lessons: why queries that look trivial are the ones that must never fail and how we measure them with an LLM-as-judge framework; why perceived difficulty, not model capability, must set the latency and cost budget; and what this forces in engineering—speculative execution, predictable degradation, catalog-grounded safety, and small fine-tuned understanding models that beat a larger general LLM. Finally, the query distribution is non-stationary: we optimize for the demand we will have, not yesterday's logs.

CCS CONCEPTS

• **Information systems** → **Retrieval models and ranking**; *Query representation*; • **Computing methodologies** → *Natural language processing*.

KEYWORDS

generative search; retrieval-augmented generation; query understanding; production systems; LLM-as-judge evaluation; nDCG; conversational search; trust and safety; distribution shift

ACM Reference Format:
Jitesh Mehta, Rahul Joshi, and Adarshan Sivashunmugam. 2026. From Demo to Production: Generative Search Lessons from Alexa+. In *Proceedings of the 35th ACM International Conference on Information and Knowledge Management (CIKM '26), November 7–11, 2026, Rome, Italy*. ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3799682.3839252

1 INTRODUCTION: THE HIDDEN SPECIFICATION

This work was built for the launch of Alexa+ on Fire TV and Echo Show, where search is a voice-first, living-room interaction serving millions of customers. Layering generative AI onto it meets constraints no benchmark captures: near-zero tolerance for failure on a voice surface where customers rarely click, a 300–500 ms responsiveness threshold [8] against multi-second LLM latency, and per-query cost that scales with traffic. Our central finding is that perceived difficulty—how hard a request looks—sets both the quality and the latency customers expect, and the two are not equally negotiable: latency can degrade gracefully under stress, the relevance and safety floor cannot.

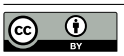
2 SYSTEM: PERCEIVED VS. STRUCTURAL DIFFICULTY

The system routes on a cheap, understanding-time estimate of perceived difficulty [3, 9]. A QU (query-understanding) stage normalizes the query and entity-tags it (title, person, genre, sort_type), then asks whether those mentions look like attributes our structured retrievers can satisfy. That surface estimate usually predicts structural difficulty—whether retrieval can actually serve the query—but not always, and the misses are what hurt. "New Dean Potter documentary" tags cleanly as person+genre+sort_type, so QU reads a simple lookup; structurally it is hard, because the person is the film's subject, not cast—a relation no structured field captures, surfaced only at retrieval. That is the dangerous tail: perceived-simple but structurally hard. QU gives a cheap estimate of how hard a query looks, never a guarantee of whether you can serve it—so you must engineer for the gap.

That gap is why we never bet on the routing estimate alone: we exploit *speculative execution*. On every query the fast attribute path and the cheaper semantic fallback [2] run in parallel. If the fast path returns a confident result it is served and the speculative results discarded; if it falls short, the fallback is already in hand at no added latency. When a query genuinely needs the LLM tier, those same results seed its retrieval-augmented generation (RAG) [1] context—added cost, but no loss of relevance. A kNN lookup costs orders of magnitude less than generative tokens, so we pay the modest cost to guarantee a fallback is always in hand—no dead ends. A request-level cache absorbs the repeated structured head; beyond it, requests escalate [4]—attribute, semantic, hybrid, then LLM—only as far as needed.

3 LESSON 1: PERCEIVED QUALITY BEATS TRAFFIC-WEIGHTED METRICS

The misses we most fear are the ones customers consider trivial: a title lookup feels easy, yet voice breaks it in several ways. Automatic speech recognition (ASR) introduces phonetic variants ("Life is Rough" for *Life is Ruff*) and mic-cutoff truncations ("Catch me if you" for *Catch Me If You Can*), while customers themselves misremember titles, naming a plausible real one ("Jurassic Park



CIKM '26, November 7–11, 2026, Rome, Italy

ACM ISBN 979-8-4007-2539-5/2026/11.
https://doi.org/10.1145/3799682.3839252

Table 1: Tail-weighted nDCG@5: latency-optimized baseline vs. +graceful fallback (both Alexa+).

Eval stratum	Quality bar	Base	+Fb
Simple attribute	near-perfect	0.83	0.83
Surface-distorted (ASR)	near-perfect	0.50	0.71
Misremembered title	near-perfect	0.58	0.65
Deep-reasoning (tone, plot)	tolerant	0.53	0.53
Safety-sensitive (maturity)	zero-tol.	0.60	0.90

Rebirth" for *Jurassic World Rebirth*)—the hardest case, because exact matching confidently returns the wrong, real title and nothing flags it. Each is "I just asked for a show by name," so a miss is maximally damaging; the path needs phonetic, fuzzy, and partial-query robustness, plus recency- and popularity-aware disambiguation. Traffic-weighted metrics tolerate these low-volume failures, and neither exact-match nor click signals can recognize success on a misremembered title; an LLM judge can. We therefore judge relevance with an LLM-as-judge rubric [5] reported as normalized discounted cumulative gain (nDCG) [6], over sets stratified by the challenge a query poses, so the looks-easy tail is over-represented and regressions surface before launch (Table 1).

We validated this judge against three domain experts on 281 queries (three-grade scale): per-rater weighted Cohen's κ of 0.95, 0.88, and 0.87 places the LLM within the human–human agreement range (0.83–0.88), with Fleiss' $\kappa = 0.83$ overall. Against a latency-optimized baseline, the full pipeline lifts nDCG@5 on the looks-easy tail from 0.51 to 0.70, with no regression on the structured head.

4 LESSON 2: PERCEIVED DIFFICULTY SETS BOTH LATENCY AND ECONOMICS

Perceived difficulty also sets the budget, and honoring it settles the economics. Invoking an LLM on every query is too slow for the living room and, at this scale, economic suicide. Keeping the structured majority on near-zero-cost attribute retrieval makes the experience viable: those queries return in the 200–300 ms range and several orders of magnitude cheaper than the LLM path, with no relevance loss, since deterministic matching avoids the LLM's semantic false positives. The lesson: spend only the compute a query's perceived difficulty warrants, not a bigger model.

5 LESSON 3: DEGRADE PREDICTABLY, OR NOT AT ALL

On a voice surface the tail is existential: p95/p99, not the mean, are what customers feel [10], and a single slow or unheard turn trains them to abandon voice altogether—a rare failure costs the modality, not one query. Traffic surges (daily, seasonal, and live-event spikes) we shed—timeouts, per-index circuit breakers, and bounded concurrency that rejects rather than queues. LLM-tier throttling we cannot autoscale, so we route around it: because the retrieval tiers always run speculatively (§2), a throttled or slow LLM degrades to a retrieval answer already in hand, at no added latency. Degradation follows the relevance axis—a failed factual query falls back to semantic retrieval, never to a lexical path that returns confident garbage and violates the looks-easy-tail bar (Lesson 1).

6 LESSON 4: TRUST AND SAFETY IS NOT OPTIONAL IN THE LIVING ROOM

A generative system serving families must neither invent content nor surface inappropriate results; two mechanisms carry the weight. First, catalog-grounded generation: the model never free-generates titles; it reformulates intent and ranks over candidates retrieved from the Fire TV catalog, structurally preventing hallucinated titles [7]. Second, safety as a retrieval constraint: maturity and parental-control requirements are hard filters on the candidate set, so a non-deterministic model can never override a safety decision. Because customers rarely switch to a kids profile, these constraints must be inferred from the session itself—"recommend something to watch with my 12-year-old" has to trigger age-appropriate filtering even on an adult account.

7 LESSON 5: ON THE UNDERSTANDING LAYER, SMALL FINE-TUNED MODELS BEAT BIG GENERAL ONES

Query understanding and entity tagging—which decide perceived difficulty—are where we expected a frontier LLM to help most, and it helped least. Domain-tuned small language and named-entity-recognition models beat a much larger general LLM at entity tagging by roughly 20% on accuracy while being faster, cheaper, and—decisively—consistent. A general LLM is brittle: prompted identically, it intermittently mis-tags obvious entities—tagging "Marvel" as a title rather than a franchise—while the fine-tuned model returns the same correct tag every time. We track this as *tag stability*—label agreement across repeated runs and paraphrases—where the fine-tuned model reached 99.5% against 94.6% for the general LLM. At the understanding layer, invest in small specialized models, not a bigger general one.

8 LESSON 6: THE QUERY DISTRIBUTION IS NON-STATIONARY

Complex queries such as "movies for date-night?"—under 1% of Fire TV traffic two years ago—are now near 6% and climbing as Alexa+'s conversational ability improves. This non-stationarity traps evaluation: metrics scored on yesterday's traffic under-weight the very tail we aim to win, quietly greenlighting the status quo. So we treat the emerging tail as a leading indicator—mining queries that escalate to the LLM or score low under the judge—and seed evaluation sets from capabilities we are about to ship, not from past logs. Perceived difficulty itself moves: as customers learn to ask what the assistant can finally answer, the easy/hard boundary the system is built on drifts. We optimize for the distribution we will have, not the one we measured.

9 CONCLUSION

In production, generative search is less about model size than about the gap between how hard a query looks and how hard it is to serve: estimate perceived difficulty cheaply, favor small specialized models, and let latency and cost flex to that estimate—but never the relevance and safety floor, least of all on the queries that look easy.

SPEAKER BIO

Jitesh Mehta is a Senior Software Development Manager at Amazon, responsible for Alexa+ and the video search and discovery experience across Amazon devices including Fire TV and Echo Show, serving millions of customers. His work centers on the cost, latency, and reliability of production LLM-powered retrieval in the living room; he presents it with co-authors Rahul Joshi and Adarshan Sivashunmugam.

GENAI USAGE DISCLOSURE

Generative AI tools were used in three ways; all outputs were reviewed and verified by the authors, who take full responsibility for the content. Writing: an AI assistant (Kiro) helped condense and copy-edit the manuscript to the page limit. Development: AI coding assistants (Kiro with Claude) assisted software development while building the production system described here. Evaluation: relevance is scored by an LLM-as-judge, validated against three human domain experts as described in Lesson 1. All lessons, data, and results derive from the real production system.

REFERENCES

[1] P. Lewis et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. *NeurIPS*, 2020.
[2] P. Nigam et al. Semantic product search. *KDD*, 2019.
[3] I. Ong et al. RouteLLM: Learning to route LLMs with preference data. *arXiv:2406.18665*, 2024.
[4] J. Arguello et al. Sources of evidence for vertical selection. *SIGIR*, 2009.
[5] L. Zheng et al. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. *NeurIPS*, 2023.
[6] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM TOIS*, 2002.
[7] K. Shuster et al. Retrieval augmentation reduces hallucination in conversation. *EMNLP Findings*, 2021.
[8] J. Nielsen. *Usability Engineering*. Morgan Kaufmann, 1993.
[9] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. Predicting query performance. *SIGIR*, 2002.
[10] J. Dean and L. A. Barroso. The tail at scale. *Commun. ACM*, 56(2):74–80, 2013.